UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA

Sam M. Gibbons U.S. Courthouse
801 North Florida Avenue
Tampa, FL 33602

Sheryl L. Loesch
Clerk

Mark Middlebrook
Tampa Division Manager

DATE: August 26, 2015

TO: Clerk, U.S. Court of Appeals for the Federal Circuit

FAIRWARNING IP, LLC,

Plaintiff,

v.

Case No: 8:14-cv-2685-T-23MAP

IATRIC SYSTEMS, INC.,

Defendant.

U.S.C.A. Case No.: ***USCA Case Number ***

Enclosed are documents and information relating to an appeal in the above-referenced action. Please acknowledge receipt on the enclosed copy of this letter.

- Honorable Steven D. Merryday, Chief United States District Judge appealed from.
- Appeal filing fee was paid.
- Certified copy of Notice of Appeal, docket entries, judgment and/or Order appealed from. Opinion was not entered orally.
- No hearing from which a transcript could be made.

SHERYL L. LOESCH, CLERK

By: s/E. Calderon, Deputy Clerk

UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA
TAMPA DIVISION

FAIRWARNING IP, LLC

Plaintiff;

v.

Case No. 8:14-cv-2685-T-23MAP

IATRIC SYSTEMS, INC.

Defendant.

PLAINTIFF'S NOTICE OF APPEAL

Notice is hereby given that Plaintiff FairWarning IP, LLC appeals to the United States Court of Appeals for the Federal Circuit from the Judgment in a Civil Case of this Court entered on July 16, 2015, (ECF 62), and from any and all other judgments, orders, opinions, rulings, and findings pertinent or ancillary to the foregoing, including without limitation the Court's Order of June 24, 2015 (ECF 56) and the Court's Order of July 15, 2015 (ECF 61).

Payment of the required fee of $505, representing the $500 fee for docketing a case on appeal specified in 28 U.S.C. §1913, and the $5 fee for filing a notice of appeal specified in 28 U.S.C. §1917, will be provided under separate cover.

Dated: August 12, 2015

Respectfully Submitted,

/s/ Jason P. Stearns
Jason P. Stearns
Florida Bar No. 059550
Michael S. Hooker
Florida Bar No. 330655

PHELPS DUNBAR LLP
100 South Ashley Drive, Suite 1900
Tampa, FL 33602
Phone: (813) 472-7550
Fax: (813) 472-7570
Email: Michael.Hooker@phelps.com
E-mail: Jason.Stearns@phelps.com

Sean Passino
(admitted *pro hac vice*)
Rachel Pilloff
(admitted *pro hac vice*)
LOWE HAUPTMAN & HAM, LLP
2318 Mill Road, Suite 1400
Alexandria, VA 22314
Phone: (703) 684-1111
Fax: (703) 518-5499

Counsel for FairWarning IP, LLC

CERTIFICATE OF SERVICE

I hereby certify that on August 12, 2015, a true and correct copy of the foregoing was filed with the Court through the ECF system and that a copy will be electronically served on registered participants as identified on the Notice of Electronic Filing.

/s/ Jason P. Stearns
Attorney

APPEAL, CLOSED

U.S. District Court
Middle District of Florida (Tampa)
CIVIL DOCKET FOR CASE #: 8:14-cv-02685-SDM-MAP

FairWarning IP, LLC v. Iatric Systems, Inc.
Assigned to: Judge Steven D. Merryday
Referred to: Magistrate Judge Mark A. Pizzo
Cause: 35:271 Patent Infringement

Date Filed: 10/24/2014
Date Terminated: 07/16/2015
Jury Demand: Plaintiff
Nature of Suit: 830 Patent
Jurisdiction: Federal Question

Plaintiff

FairWarning IP, LLC

represented by **Michael S. Hooker**
Phelps Dunbar, LLP
Suite 1900
100 S Ashley Dr
Tampa, FL 33602-5311
813/472-7550
Fax: 813/472-7570
Email: hookerm@phelps.com
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Jason Paul Stearns
Phelps Dunbar, LLP
Suite 1900
100 S Ashley Dr
Tampa, FL 33602-5311
813/472-7550
Fax: 813/472-7550
Email: stearnsj@phelps.com
ATTORNEY TO BE NOTICED

Lawrence Joseph Dougherty
Foley &Lardner, LLP
Suite 2700
100 N Tampa St
Tampa, FL 33602
813 229-2300
Fax: 813 225-4210
Email: ldougherty@foley.com
TERMINATED: 07/01/2015

Rachel Karen Pilloff
Lowe Hauptman &Ham, LLP
2318 Mill Rd Ste 1400
Alexandria, VA 22314-6878
703-373-4400
Fax: 703-518-5499
Email: rpilloff@ipfirm.com
ATTORNEY TO BE NOTICED

Robert J. Silverman
Foley &Lardner, LLP
111 Huntington Ave
Boston, MA 02199
617/342-4000
Fax: 617/342-4001
Email: rsilverman@foley.com
TERMINATED: 07/01/2015
PRO HAC VICE

Sean Allen Passino
Lowe Hauptman &Ham, LLP
2318 Mill Rd Ste 1400
Alexandria, VA 22314-6878
703-535-6525
Fax: 703-518-5499
Email: spassino@ipfirm.com
ATTORNEY TO BE NOTICED

Thomas I. Elkind
Foley &Lardner, LLP
111 Huntington Ave
Boston, MA 02199
617-342-4000
Fax: 617-342-4001
Email: telkind@foley.com
TERMINATED: 07/01/2015

V.

Defendant

Iatric Systems, Inc.

represented by

Brandon Scruggs
Sunstein Kann Murphy &Timbers LLP
125 Summer St
Boston, MA 02110
617-443-9292
Fax: 617-443-0004
Email: bscruggs@sunsteinlaw.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Lisa M. Tittemore
Sunstein Kann Murphy &Timbers LLP
125 Summer St
Boston, MA 02110
617-443-9292
Email: ltittemore@sunsteinlaw.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Sharona H. Sternberg
Sunstein Kann Murphy &Timbers LLP
125 Summer St
Boston, MA 02110
617-443-9292
Email: ssternberg@sunsteinlaw.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Catherine Fly Yant
Fee &Jeffries PA
1227 N Franklin St
Tampa, FL 33602-8002
813/229-8008
Email: cyant@feejeffries.com
ATTORNEY TO BE NOTICED

Kathleen M. Wade
Fee &Jeffries PA
1227 N Franklin St
Tampa, FL 33602-8002

813.229.8008
Fax: 813.229.0046
Email: kwade@feejeffries.com
ATTORNEY TO BE NOTICED

Richard Edson Fee
Fee &Jeffries PA
1227 N Franklin St
Tampa, FL 33602-8002
813/229-8008
Fax: 813/229-0046
Email: rfee@feejeffries.com
ATTORNEY TO BE NOTICED

Date Filed	#	Docket Text
10/24/2014	1	COMPLAINT against Iatric Systems, Inc. with Jury Demand (Filing fee $ 400 receipt number TPA-26407) filed by FairWarning IP, LLC. (Attachments: #1 Civil Cover Sheet, #2 Exhibit A)(LSS) (Entered: 10/27/2014)
10/24/2014	2	NOTICE of pendency of related cases per Local Rule 1.04(d) by FairWarning IP, LLC. Related case(s): yes (LSS) (Entered: 10/27/2014)
10/24/2014	3	Patent Report sent to Alexandria, VA. (LSS) (Entered: 10/27/2014)
10/24/2014	4	SUMMONS issued as to Iatric Systems, Inc.. (LSS) (Entered: 10/27/2014)
10/30/2014	5	MOTION for Robert J. Silverman to appear pro hac vice by FairWarning IP, LLC. (Dougherty, Lawrence) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 10/30/2014)
10/30/2014	6	MOTION for Thomas I. Elkind to appear pro hac vice by FairWarning IP, LLC. (Dougherty, Lawrence) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 10/30/2014)
10/30/2014	7	MOTION for Sean A. Passino to appear pro hac vice by FairWarning IP, LLC. (Dougherty, Lawrence) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 10/30/2014)
10/30/2014	8	MOTION for Rachel K. Pilloff to appear pro hac vice by FairWarning IP, LLC. (Dougherty, Lawrence) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 10/30/2014)
10/31/2014	9	ORDER granting 5 motion for Robert J. Silverman to appear pro hac vice; granting 6 motion for Thomas I. Elkind to appear pro hac vice; granting 7 motion for Sean A. Passino to appear pro hac vice; granting 8 motion for Rachel K. Pilloff to appear pro hac vice. Signed by Magistrate Judge Mark A. Pizzo on 10/31/2014. (SSW) (Entered: 10/31/2014)
11/03/2014	10	DOCUMENT TERMINATED counsel notified to refile using the correct event code. NOTICE by FairWarning IP, LLC re 2 Notice of pendency of related cases, 1 Complaint *Notice of Filing Affidavit of Service* (Dougherty, Lawrence) Modified on 11/4/2014 (EJC). (Entered: 11/03/2014)
11/03/2014		(Court only) ***COPIES mailed to Counsel: Robert J. Silverman re 9 Order on motion to appear pro hac vice (EJC) (Entered: 11/03/2014)
11/04/2014		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Sean Allen Passino, appearing on behalf of FairWarning IP, LLC (Filing fee $10 receipt number TPA26547.). (JNB) (Entered: 11/04/2014)
11/04/2014		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Rachel Karen Pilloff, appearing on behalf of FairWarning IP, LLC (Filing fee $10 receipt number TPA26547.). (JNB) (Entered: 11/04/2014)
11/04/2014		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Robert J. Silverman, appearing on behalf of FairWarning IP, LLC (Filing fee $10 receipt number TPA26547.). (JNB) (Entered: 11/04/2014)

11/04/2014		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Thomas I. Elkind, appearing on behalf of FairWarning IP, LLC (Filing fee $10 receipt number TPA26547.). (JNB) (Entered: 11/04/2014)
11/06/2014	11	RETURN of service executed on October 24, 2014 by FairWarning IP, LLC as to Iatric Systems, Inc.. (Dougherty, Lawrence) (Entered: 11/06/2014)
11/06/2014	12	NOTICE of compliance re 9 Order on motion to appear pro hac vice *as to Robert J. Silverman* by FairWarning IP, LLC (Silverman, Robert) (Entered: 11/06/2014)
11/07/2014	13	NOTICE of compliance by FairWarning IP, LLC (Passino, Sean) (Entered: 11/07/2014)
11/07/2014	14	NOTICE of compliance by FairWarning IP, LLC (Pilloff, Rachel) (Entered: 11/07/2014)
11/07/2014	15	DOCUMENT TERMINATED counsel notified to refile using the correct event code. Unopposed MOTION for Extension of Time to File Response/Reply as to 1 Complaint by Iatric Systems, Inc.. (Fee, Richard) Motions referred to Magistrate Judge Mark A. Pizzo. Modified on 11/10/2014 (EJC). (Entered: 11/07/2014)
11/10/2014	16	Unopposed MOTION for Extension of Time to File Answer re 1 Complaint *[filing per Clerk's instruction to correct event description]* by Iatric Systems, Inc.. (Fee, Richard) (Entered: 11/10/2014)
11/10/2014	17	NOTICE of compliance re 9 Order on motion to appear pro hac vice by FairWarning IP, LLC (Elkind, Thomas) (Entered: 11/10/2014)
11/12/2014		(Court only) *** 16 Unopposed MOTION for Extension of Time to File Answer re 1 Complaint *[filing per Clerk's instruction to correct event description]* REFERRED to Magistrate Judge Mark A. Pizzo. (EJC) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 11/12/2014)
11/12/2014	18	ENDORSED ORDER granting 16 Motion for Extension of Time to Answer or respond to Complaint. Iatric Systems, Inc.'s answer or response to Plaintiff's complaint is due 12/15/2014. Signed by Magistrate Judge Mark A. Pizzo on 11/12/2014. (SSW) (Entered: 11/12/2014)
11/21/2014	19	NOTICE of designation under Local Rule 3.05 – Track 2 (TKD) (Entered: 11/21/2014)
12/05/2014	20	Unopposed MOTION for Lisa M. Tittemore to appear pro hac vice by Iatric Systems, Inc.. (Fee, Richard) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 12/05/2014)
12/05/2014	21	Unopposed MOTION for Brandon Scruggs to appear pro hac vice by Iatric Systems, Inc.. (Fee, Richard) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 12/05/2014)
12/08/2014	22	ORDER granting 20 motion for Lisa D. Tittemore to appear pro hac vice; granting 21 motion for Brandon Scruggs to appear pro hac vice. Signed by Magistrate Judge Mark A. Pizzo on 12/8/2014. (SSW) (Entered: 12/08/2014)
12/09/2014		(Court only) ***Attorney Lisa M. Tittemore for Iatric Systems, Inc. added. (EJC) (Entered: 12/09/2014)
12/09/2014		(Court only) ***Attorney Brandon Scruggs for Iatric Systems, Inc. added. (EJC) (Entered: 12/09/2014)
12/12/2014		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Lisa M. Tittemore, appearing on behalf of Iatric Systems, Inc. (Filing fee $10 receipt number TPA027359.) Related document: 20 Unopposed MOTION for Lisa M. Tittemore to appear pro hac vice. (AG) (Entered: 12/15/2014)
12/12/2014		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Brandon Scruggs, appearing on behalf of Iatric Systems, Inc. (Filing fee $10 receipt number TPA027359.) Related document: 21 Unopposed MOTION for Brandon Scruggs to appear pro hac vice. (AG) (Entered: 12/15/2014)

12/15/2014	23	MOTION to dismiss for failure to state a claim *and Request for Oral Argument* by Iatric Systems, Inc.. (Fee, Richard) (Entered: 12/15/2014)
12/16/2014		(Court only) ***Deadlines terminated. Answer deadline cancelled. (JNB) (Entered: 12/16/2014)
12/18/2014	24	NOTICE of compliance *with Court Order* by Iatric Systems, Inc. (Tittemore, Lisa) (Entered: 12/18/2014)
12/18/2014	25	NOTICE of compliance *with Court Order* by Iatric Systems, Inc. (Scruggs, Brandon) (Entered: 12/18/2014)
12/24/2014	26	DOCUMENT TERMINATED counsel notified to refile using the correct attorney log in and signature. Unopposed MOTION for Extension of Time to File Response/Reply as to 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument* by FairWarning IP, LLC. (Dougherty, Lawrence) Motions referred to Magistrate Judge Mark A. Pizzo. Modified on 12/29/2014 (EJC). (Entered: 12/24/2014)
12/29/2014		(Court only) ***Motions terminated: 26 Unopposed MOTION for Extension of Time to File Response/Reply as to 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument* filed by FairWarning IP, LLC. (EJC) (Entered: 12/29/2014)
12/29/2014	27	Unopposed MOTION for Extension of Time to File Response/Reply as to 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument – Renewed Agreed Motion for Extension of Time for Plaintiff FairWarning IP, LLC to File Its Response in Opposition to Defendant's Motion to Dismiss* by FairWarning IP, LLC. (Silverman, Robert) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 12/29/2014)
12/30/2014		(Court only) ***Motions no longer referred: 27 Unopposed MOTION for Extension of Time to File Response/Reply as to 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument – Renewed Agreed Motion for Extension of Time for Plaintiff FairWarning IP, LLC to Fi. (RFK) (Entered: 12/30/2014)*
01/06/2015	28	ENDORSED ORDER granting 27 the plaintiff's motion for an extension through January 16, 2015, of the time within which to respond to the motion to dismiss. Signed by Judge Steven D. Merryday on 1/6/2015. (Entered: 01/06/2015)
01/06/2015	29	CASE MANAGEMENT REPORT. (Attachments: #1 Exhibit A)(Dougherty, Lawrence) (Entered: 01/06/2015)
01/07/2015		(Court only) Set/Reset Deadlines as to 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument*. Responses due by 1/16/2015 (EJC) (Entered: 01/07/2015)
01/07/2015	30	NOTICE of Appearance by Kathleen M. Wade on behalf of Iatric Systems, Inc. (Wade, Kathleen) (Entered: 01/07/2015)
01/16/2015	31	RESPONSE in Opposition re 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument* filed by FairWarning IP, LLC. (Attachments: #1 Affidavit Kurt Long Declaration)(Dougherty, Lawrence) (Entered: 01/16/2015)
01/17/2015	32	NOTICE by FairWarning IP, LLC re 31 Response in Opposition to Motion (Attachments: #1 Exhibit A to Long Declaration, #2 Exhibit B to Long Declaration, #3 Exhibit C to Long Declaration, #4 Exhibit D to Long Declaration)(Dougherty, Lawrence) (Entered: 01/17/2015)
01/17/2015	33	NOTICE by FairWarning IP, LLC re 31 Response in Opposition to Motion (Attachments: #1 Affidavit Declaration of Lawrence J. Dougherty, #2 Exhibit A to Dougherty Declaration, #3 Exhibit B to Dougherty Declaration, #4 Exhibit C–1 to Dougherty Declaration, #5 Exhibit C–2 to Dougherty Declaration, #6 Exhibit D to Dougherty Declaration)(Dougherty, Lawrence) (Entered: 01/17/2015)
01/20/2015		(Court only) ***Deadlines terminated. Past due deadlines cancelled. (JNB) (Entered: 01/20/2015)

01/20/2015	34	AMENDED RESPONSE in Opposition re 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument* filed by FairWarning IP, LLC. (Attachments: # 1 Exhibit Declaration of Kurt J. Long)(Dougherty, Lawrence) Modified on 1/20/2015 (EJC) (Entered: 01/20/2015)
01/27/2015	35	CASE MANAGEMENT AND SCHEDULING ORDER: Pretrial Conference scheduled for 11/8/2016 at 10:00 AM in Tampa Courtroom 11B before Magistrate Judge Mark A. Pizzo; bench trial scheduled for January 2017 trial calendar.See order for other deadlines. Signed by Judge Steven D. Merryday on 1/27/2015. (BK) (Entered: 01/27/2015)
01/28/2015	36	DEMAND for trial by jury by FairWarning IP, LLC. (Dougherty, Lawrence) (Entered: 01/28/2015)
02/03/2015	37	AMENDED CASE MANAGEMENT AND SCHEDULING ORDER: Jury trial scheduled for the January 2017 trial calendar in Tampa Courtroom 15A before Judge Steven D. Merryday. Signed by Judge Steven D. Merryday on 2/3/2015. (BK) (Entered: 02/03/2015)
02/10/2015	38	NOTICE of pendency of related cases per Local Rule 1.04(d) by FairWarning IP, LLC. Related case(s): yes (Dougherty, Lawrence) (Entered: 02/10/2015)
02/12/2015	39	NOTICE of supplemental authority re 23 MOTION to dismiss for failure to state a claim *and Request for Oral Argument* by Iatric Systems, Inc.. (Attachments: # 1 Exhibit A, # 2 Exhibit B, # 3 Exhibit C)(Fee, Richard) (Entered: 02/12/2015)
02/17/2015	40	MOTION for leave to file Response to Defendant's Notice of Supplemental Authority (Doc 39) by FairWarning IP, LLC. (Dougherty, Lawrence) (Entered: 02/17/2015)
02/18/2015	41	ENDORSED ORDER denying 40 Fair Warning IP's motion for leave to respond to a notice of supplemental authority. Signed by Judge Steven D. Merryday on 2/18/2015. (Entered: 02/18/2015)
03/03/2015	42	NOTICE of unavailability of counsel by Iatric Systems, Inc. from 03/19/2015 to 03/27/2015. (Fee, Richard) (Entered: 03/03/2015)
03/10/2015	43	ORDER construing Doc. 23 as a motion for more definite statement; granting 23 --motion for more definite statement; directing the plaintiff to amend the complaint by 3/18/2015. Signed by Judge Steven D. Merryday on 3/10/2015. (BK) (Entered: 03/10/2015)
03/10/2015		(Court only) *** Amended complaint due 3/18/2015 (BK) (Entered: 03/10/2015)
03/13/2015	44	Unopposed MOTION for Sharona H. Sternberg to appear pro hac vice by Iatric Systems, Inc.. (Wade, Kathleen) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 03/13/2015)
03/16/2015	45	ORDER granting 44 motion for Sharona H. Sternberg to appear pro hac vice. Signed by Magistrate Judge Mark A. Pizzo on 3/16/2015. (SSW) (Entered: 03/16/2015)
03/16/2015		(Court only) ***Attorney Sharona H. Sternberg for Iatric Systems, Inc. added. (EJC) (Entered: 03/16/2015)
03/16/2015		(Court only) ***COPIES mailed to Counsel: Sharona H. Sternberg re 45 Order on motion to appear pro hac vice (EJC) (Entered: 03/16/2015)
03/18/2015		***PRO HAC VICE FEES paid and Special Admission Attorney Certification Form filed by attorney Sharona H. Sternberg, appearing on behalf of Iatric Systems, Inc. (Filing fee $150 receipt number TPA029054.) Related document: 44 Unopposed MOTION for Sharona H. Sternberg to appear pro hac vice. (AG) (Entered: 03/18/2015)
03/18/2015	46	DOCUMENT TERMINATED counsel refiled using the correct attorney log in and signature. AMENDED COMPLAINT *pursuant to the Court's Order ECF 43* against Iatric Systems, Inc. with Jury Demand. filed by FairWarning IP, LLC. Related document: 1 Complaint filed by FairWarning IP, LLC. (Attachments: # 1 Exhibit A 8,578,500 Patent)(Dougherty, Lawrence) Modified on 3/19/2015 (EJC).

		(Entered: 03/18/2015)
03/18/2015	47	AMENDED COMPLAINT *pursuant to the Court's Order ECF 43* against Iatric Systems, Inc. with Jury Demand. filed by FairWarning IP, LLC. Related document: 1 Complaint filed by FairWarning IP, LLC. (Attachments: #1 Exhibit A U.S. Patent 8,578,500)(Dougherty, Lawrence) (Entered: 03/18/2015)
03/19/2015		(Court only) ***Deadlines terminated. Past due deadlines cancelled. (JNB) (Entered: 03/19/2015)
03/19/2015	48	Patent Report sent to Alexandria, VA. (EJC) (Entered: 03/19/2015)
03/24/2015	49	NOTICE of compliance re 45 Order on motion to appear pro hac vice by Iatric Systems, Inc. (Sternberg, Sharona) (Entered: 03/24/2015)
04/06/2015	50	MOTION to dismiss for failure to state a claim *and Request for Oral Argument* by Iatric Systems, Inc.. (Attachments: #1 Exhibit A, #2 Exhibit B, #3 Exhibit C, #4 Exhibit D)(Fee, Richard) (Entered: 04/06/2015)
04/20/2015	51	NOTICE by Iatric Systems, Inc. *of Filing of Motion for Consolidation in Related Case* (Attachments: #1 Exhibit A – Copy of motion to consolidate)(Fee, Richard) (Entered: 04/20/2015)
04/23/2015	52	RESPONSE re 50 MOTION to dismiss for failure to state a claim *and Request for Oral Argument* by FairWarning IP, LLC. (Dougherty, Lawrence) (Entered: 04/23/2015)
04/23/2015	53	DECLARATION of Kurt Long re 52 Response by FairWarning IP, LLC. (Attachments: #1 Exhibit A, #2 Exhibit B, #3 Exhibit C, #4 Exhibit D)(Dougherty, Lawrence) (Entered: 04/23/2015)
04/23/2015	54	DECLARATION of Larry Dougherty re 52 Response by FairWarning IP, LLC. (Attachments: #1 Exhibit A–1, #2 Exhibit A–2, #3 Exhibit B, #4 Exhibit C, #5 Exhibit D, #6 Exhibit E, #7 Exhibit F, #8 Exhibit G, #9 Exhibit H–1, #10 Exhibit H2)(Dougherty, Lawrence) (Entered: 04/23/2015)
06/23/2015		(Court only) ***Deadlines terminated. Past due deadlines cancelled. (JNB) (Entered: 06/23/2015)
06/24/2015	55	NOTICE of Appearance by Catherine Fly Yant on behalf of Iatric Systems, Inc. (Yant, Catherine) (Entered: 06/24/2015)
06/24/2015	56	**ORDER denying 50 —motion for oral argument; granting 50 —motion to dismiss; dismissing the complaint without prejudice; amended complaint due 7/9/2015. Signed by Judge Steven D. Merryday on 6/24/2015. (BK)** (Entered: 06/24/2015)
06/24/2015		(Court only) ***Set amended complaint deadline to 7/9/2015. (BK) (Entered: 06/24/2015)
06/24/2015	57	NOTICE of Appearance by Michael S. Hooker on behalf of FairWarning IP, LLC (Hooker, Michael) (Entered: 06/24/2015)
06/24/2015	58	NOTICE of Appearance by Jason Paul Stearns on behalf of FairWarning IP, LLC (Stearns, Jason) (Entered: 06/24/2015)
06/30/2015	59	Unopposed MOTION for Lawrence J. Dougherty, Robert J. Silverman and Thomas I. Elkind to withdraw as attorney by FairWarning IP, LLC. (Dougherty, Lawrence) (Entered: 06/30/2015)
07/01/2015		(Court only) *** 59 Unopposed MOTION for Lawrence J. Dougherty, Robert J. Silverman and Thomas I. Elkind to withdraw as attorney REFERRED to Magistrate Judge Mark A. Pizzo. (EJC) Motions referred to Magistrate Judge Mark A. Pizzo. (Entered: 07/01/2015)
07/01/2015	60	**ENDORSED ORDER granting 59 Motions of Attorneys Robert J. Silverman, Lawrence Joseph Dougherty, and Thomas I. Elkind, as counsel for FairWarning IP, LLC. Signed by Magistrate Judge Mark A. Pizzo on 7/1/2015. (Pizzo, Mark)** (Entered: 07/01/2015)

07/10/2015		(Court only) ***Deadlines terminated. Past due deadlines cancelled. (JNB) (Entered: 07/10/2015)
07/15/2015	61	**ORDER dismissing the action WITH PREJUDICE. The clerk is directed to enter for the defendant and against the plaintiff a judgment dismissing this action. Further, the clerk is directed to terminate any pending motion and to close the case. Signed by Judge Steven D. Merryday on 7/15/2015. (LAM)** (Entered: 07/15/2015)
07/16/2015	62	**CLERK'S JUDGMENT in favor of Iatric Systems, Inc. against FairWarning IP, LLC Signed by Deputy Clerk on 7/16/2015. CASE CLOSED.(EJC)** (Entered: 07/16/2015)
07/16/2015	63	Patent Report sent to Alexandria, VA. (EJC) (Entered: 07/16/2015)
07/16/2015		(Court only) ***Clear Trial Set flag (EJC) (Entered: 07/16/2015)
08/12/2015	64	***Incorrect event – Counsel to refile document*** NOTICE by FairWarning IP, LLC re 56 Order on Motion to Dismiss for Failure to State a Claim, 62 Clerk's Judgment, 61 Order dismissing case *Plaintiff's Notice of Appeal* (Stearns, Jason) Modified on 8/12/2015 (AMD). (Entered: 08/12/2015)
08/12/2015	65	NOTICE OF APPEAL as to 56 Order on Motion to Dismiss for Failure to State a Claim, 62 Clerk's Judgment, 61 Order dismissing case by FairWarning IP, LLC. Filing fee not paid. (Stearns, Jason) (Entered: 08/12/2015)
08/12/2015	66	NOTICE by FairWarning IP, LLC re 65 Notice of appeal *Plaintiffs Notice of Striking Docket Entry 64* (Stearns, Jason) (Entered: 08/12/2015)
08/14/2015		USCA appeal fees received $ 505 receipt number tpa31687 re 65 Notice of appeal filed by FairWarning IP, LLC (ARC) (Entered: 08/14/2015)
08/26/2015		TRANSMITTAL to USCA for the FEDERAL CIRCUIT forwarding 50 Motion; 56 Order; 61 Order; 62 Judgment re 65 Notice of appeal. (EJC) (Entered: 08/26/2015)

UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA
TAMPA DIVISION

FAIRWARNING IP, LLC,

Plaintiff,

v. **Case No: 8:14-cv-2685-T-23MAP**

IATRIC SYSTEMS, INC.,

Defendant.

JUDGMENT IN A CIVIL CASE

Decision by Court. This action came before the Court and a decision has been rendered.

IT IS ORDERED AND ADJUDGED that Judgment is for the Defendant Iatric Systems, Inc. and against Plaintiff FairWarning IP, LLC.

SHERYL L. LOESCH, CLERK

s/E. Calderon, Deputy Clerk

CIVIL APPEALS JURISDICTION CHECKLIST

1. **Appealable Orders**: Courts of Appeals have jurisdiction conferred and strictly limited by statute:

   (a) **Appeals from final orders pursuant to 28 U.S.C. Section 1291**: Only final orders and judgments of district courts, or final orders of bankruptcy courts which have been appealed to and fully resolved by a district court under 28 U.S.C. Section 158, generally are appealable. A final decision is one that "ends the litigation on the merits and leaves nothing for the court to do but execute the judgment." Pitney Bowes, Inc. V. Mestre, 701 F.2d 1365, 1368 (11th Cir. 1983). A magistrate judge's report and recommendation is not final and appealable until judgment thereon is entered by a district court judge. 28 U.S.C. Section 636(c).

   (b) **In cases involving multiple parties or multiple claims**, a judgment as to fewer than all parties or all claims is not a final, appealable decision unless the district court has certified the judgment for immediate review under Fed.R.Civ.P. 54(b), Williams v. Bishop, 732 F.2d 885, 885-86 (11th Cir. 1984). A judgment which resolves all issues except matters, such as attorneys' fees and costs, that are collateral to the merits, is immediately appealable. Budinich v. Becton Dickinson & Co., 486 U.S. 196, 201, 108 S. Ct. 1717, 1721-22, 100 L.Ed.2d 178 (1988); LaChance v. Duffy's Draft House, Inc., 146 F.3d 832, 837 (11th Cir. 1998).

   (c) **Appeals pursuant to 28 U.S.C. Section 1292(a)**: Appeals are permitted from orders "granting, continuing, modifying, refusing or dissolving injunctions or refusing to dissolve or modify injunctions..." and from "[i]nterlocutory decrees...determining the rights and liabilities of parties to admiralty cases in which appeals from final decrees are allowed." Interlocutory appeals from orders denying temporary restraining orders are not permitted.

   (d) **Appeals pursuant to 28 U.S.C. Section 1292(b) and Fed.R.App.P.5:** The certification specified in 28 U.S.C. Section 1292(b) must be obtained before a petition for permission to appeal is filed in the Court of Appeals. The district court's denial of a motion for certification is not itself appealable.

   (e) **Appeals pursuant to judicially created exceptions to the finality rule:** Limited exceptions are discussed in cases including, but not limited to: Cohen V. Beneficial Indus. Loan Corp., 337 U.S. 541,546,69 S.Ct. 1221, 1225-26, 93 L.Ed. 1528 (1949); Atlantic Fed. Sav. & Loan Ass'n v. Blythe Eastman Paine Webber, Inc., 890 F. 2d 371, 376 (11th Cir. 1989); Gillespie v. United States Steel Corp., 379 U.S. 148, 157, 85 S. Ct. 308, 312, 13 L.Ed.2d 199 (1964).

2. **Time for Filing:** The timely filing of a notice of appeal is mandatory and jurisdictional. Rinaldo v. Corbett, 256 F.3d 1276, 1278 (11th Cir. 2001). In civil cases, Fed.R.App.P.4(a) and (c) set the following time limits:

   (a) **Fed.R.App.P. 4(a)(1)**: A notice of appeal in compliance with the requirements set forth in Fed.R.App.P. 3 must be filed in the district court within 30 days after the entry of the order or judgment appealed from. However, if the United States or an officer or agency thereof is a party, the notice of appeal must be filed in the district court within 60 days after such entry. **THE NOTICE MUST BE RECEIVED AND FILED IN THE DISTRICT COURT NO LATER THAN THE LAST DAY OF THE APPEAL PERIOD - no additional days are provided for mailing.** Special filing provisions for inmates are discussed below.

   (b) **Fed.R.App.P. 4(a)(3)**: "If one party timely files a notice of appeal, any other party may file a notice of appeal within 14 days after the date when the first notice was filed, or within the time otherwise prescribed by this Rule 4(a), whichever period ends later."

   (c) **Fed.R.App.P.4(a)(4)**: If any party makes a timely motion in the district court under the Federal Rules of Civil Procedure of a type specified in this rule, the time for appeal for all parties runs from the date of entry of the order disposing of the last such timely filed motion.

   (d) **Fed.R.App.P.4(a)(5) and 4(a)(6)**: Under certain limited circumstances, the district court may extend the time to file a notice of appeal. Under Rule 4(a)(5), the time may be extended if a motion for an extension is filed within 30 days after expiration of the time otherwise provided to file a notice of appeal, upon a showing of excusable neglect or good cause. Under Rule 4(a)(6), the time may be extended if the district court finds upon motion that a party did not timely receive notice of the entry of the judgment or order, and that no party would be prejudiced by an extension.

   (e) **Fed.R.App.P.4(c)**: If an inmate confined to an institution files a notice of appeal in either a civil case or a criminal case, the notice of appeal is timely if it is deposited in the institution's internal mail system on or before the last day for filing. Timely filing may be shown by a declaration in compliance with 28 U.S.C. Section 1746 or a notarized statement, either of which must set forth the date of deposit and state that first-class postage has been prepaid.

3. **Format of the notice of appeal**: Form 1, Appendix of Forms to the Federal Rules of Appellate Procedure, is a suitable format. See also Fed.R.App.P. 3(c). A pro se notice of appeal must be signed by the appellant.

4. **Effect of a notice of appeal**: A district court loses jurisdiction (authority) to act after the filing of a timely notice of appeal, except for actions in aid of appellate jurisdiction or to rule on a timely motion of the type specified in Fed.R.App.P. 4(a)(4).

UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA
TAMPA DIVISION

FAIRWARNING IP, LLC,

Plaintiff,

v. CASE NO. 8:14-cv-2685-T-23MAP

IATRIC SYSTEMS, INC.,

Defendant.
_____/

ORDER

A June 24, 2015 order (Doc. 56) states, "No later than **JULY 9, 2015**, FairWarning may amend the complaint to assert a claim that is independent of the '500 patent's validity. If FairWarning fails to amend the complaint on or before July 9, 2015, an order will promptly dismiss this action with prejudice." FairWarning failed to amend the complaint. Accordingly, this action is **DISMISSED WITH PREJUDICE**. The clerk is directed to enter for the defendant and against the plaintiff a judgment dismissing this action with prejudice because United States Patent No. 8,578,500 is invalid under 35 U.S.C. § 101. Further, the clerk is directed to terminate any pending motion and to close the case

ORDERED in Tampa, Florida, on July 15, 2015.

STEVEN D. MERRYDAY
UNITED STATES DISTRICT JUDGE

UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA
TAMPA DIVISION

FAIRWARNING IP, LLC,

Plaintiff,

v. CASE NO. 8:14-cv-2685-T-23MAP

IATRIC SYSTEMS, INC.,

Defendant.
_____/

ORDER

FairWarning IP, LLC, sues (Doc. 47) Iatric Systems, Inc., for infringing United States Patent No. 8,578,500. Challenging the patent's validity, Iatric moves (Doc. 68) to dismiss under 35 U.S.C. § 101.

BACKGROUND

The '500 patent (Doc. 47-1) claims a "system and method of detecting fraud and/or misuse in a computer environment based on analyzing data." '500 patent, col. 1, ll. 15–17. Specifically, Claim 1 describes a "method of detecting improper access of a patient's protected health information . . . in a computer environment"; Claim 12 describes a "system" that implements Claim 1's method; and Claim 14 describes a "computer-readable medium" containing program code that performs Claim 1's method. '500 patent, col. 16, ll. 27–29; col. 17, l. 24; col. 18, l. 7. Also, the patent contains fourteen dependent claims (Claims 2–11, 13, and 15–17), each

of which adds a slight limitation to the method, the system, or the computer-readable medium.

According to FairWarning, "the '500 patent analyzes audit log data in order to identify potential snooping and identify theft by authorized users" of "electronic patient medical records." (Doc. 52 at 2) The '500 patent reviews each "user's activity, identity, frequency of activity, and the like," and "in appropriate circumstances a 'hit' is stored in memory and a 'notification' is provided." (Doc. 52 at 2, 11) Iatric challenges (Doc. 50) the '500 patent's validity and argues that the patent "claims the abstract idea of analyzing records of human activity to detect suspicious behavior, and its direction to 'apply it' in the computer context fails to describe an improvement to the function of a computer itself, or an improvement in another technological field." (Doc. 50 at 2)

DISCUSSION

Limiting the subject matter of a patent-eligible invention, 35 U.S.C. § 101 states, "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." Section 101 excludes from patent protection a law of nature, a natural phenomenon, and an abstract idea.

Alice Corp. v. CLS Bank International, 134 S. Ct. 2347, 2355 (2014), identifies a two-step analysis required to determine a patent's validity under Section 101:

> First, . . . determine whether the claims at issue are directed to one of those patent-ineligible concepts. If so . . . , then ask, "what else is there in the claims . . . ?" To answer that question, . . . consider the elements of each claim both individually and "as an ordered combination" to determine whether the additional elements "transform the nature of the claim" into a patent-eligible application.

Analysis under *Alice* begins by determining whether the "concept" that the patent is "directed to" or "drawn to" is a patentable concept. *Alice* considers a patent that, like the '500 patent, claims a method, a system, and a computer-readable medium. In *Alice*, 134 S. Ct. at 2352, "[t]he claims at issue relate to a computerized scheme for mitigating 'settlement risk'" through a third-party intermediary. Without "labor[ing] to delimit the precise contours of the 'abstract idea' category," *Alice*, 134 S. Ct. at 2356–57, explains that the patented "scheme" (known as "intermediated settlement") is a "fundamental" and "long prevalent" practice.

As Iatric correctly argues, the '500 patent is "directed to" or "drawn to" the concept of "analyzing records of human activity to detect suspicious behavior." (Doc. 50 at 2) Reviewing activity to detect suspicious behavior is not unique to the context of private health information, and binding precedent has invalidated patents "directed to" similar concepts. *E.g.*, *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1367 (Fed. Cir. 2011) (invalidating a patent that claimed a "method and system for detecting fraud in a credit card transaction between [a] consumer and a merchant over the Internet"); *accord Intellectual Ventures II LLC v. JP Morgan Chase & Co.*, 2015 WL 1941331, *3 (S.D.N.Y. April 28, 2015) (Hellerstein, J.) (invalidating a patent that claimed a "method for monitoring multiple computer hosts within a network for

anomalies, and alerting the various hosts of possible intrusion"); *Wireless Media Innovations, LLC v. Maher Terminals, LLC*, 2015 WL 1810378, *8 (D.N.J. April 20, 2015) (Linares, J.) (invalidating patents "directed to the . . . abstract idea[of] monitoring locations, movement, and load status of shipping containers within a container-receiving yard, and storing, reporting and communicating this information in various forms through generic computer functions"). Reviewing activity to detect suspicious behavior is a basic and well-established abstract idea.[1]

Attempting to demonstrate that the '500 patent is not "directed to" an abstract idea, FairWarning analogizes to *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1257 (Fed. Cir. 2014), which upholds a patent in which the "claims address[] the problem of retaining website visitors that, if adhering to the routine, conventional functioning of Internet hyperlink protocol, would be instantly transported away from a host's website after 'clicking' on an advertisement and activating a hyperlink." Finding that the patent comports with Section 101, the Federal Circuit stated that the patent "do[es] not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet." *DDR Holdings*, 773 F.3d at 1258. "Instead, the claimed solution is necessarily rooted in computer technology in order to overcome a problem specifically arising in the

[1] Also, Iatric argues that the '500 patent's "claims do little more than mirror [the Health Information Portability and Accountability Act] regulations" and that, therefore, the claims are "directed to" a "conventional (and indeed even required) activity in the industry." (Doc. 50 at 17, 19) Because this order invalidates the '500 patent, Iatric's argument that "[p]atenting compliance with HIPAA regulations obviously threatens to pre-empt the field" (Doc. 50 at 17) remains unresolved.

realm of computer networks." *DDR Holdings*, 773 F.3d at 1258. In other words, no "pre-Internet analog of the patent's asserted claims" exists because the problem addressed by claims is unique to "the realm of computer networks." *DDR Holdings*, 773 F.3d at 1257, 1258.

In a strained comparison, FairWarning argues that the '500 patent "provides a solution to a technological problem, namely, identifying potential snooping and identity theft by authorized users." (Doc. 52 at 10) However, *DDR Holdings* is inapposite because the '500 patent is not "necessarily rooted in computer technology." FairWarning asserts that "analyzing audit log data is not like analyzing human behavior, as audit log data examines the electronic footprint or trail of activities that are executed in a computer environment." (Doc. 52 at 7) But, as Iatric states, the '500 patent "is but a modern spin" (Doc. 50 at 16) on reviewing activity to detect suspicious behavior, an activity that existed in the "pre-Internet world."[2]

The '500 patent is "directed to" an abstract idea; therefore, the second *Alice* step applies. The second *Alice* step requires an examination of "the elements of the claim to determine whether it contains an 'inventive concept' sufficient to 'transform' the claimed abstract idea into a patent-eligible application." *Alice*, 134 S. Ct. at 2357. A successful transformation requires "more than simply stating the abstract idea while adding the words 'apply it.'" *Alice*, 134 S. Ct. at 2357.

---

[2] Even if FairWarning could identify a meaningful distinction between reviewing "audit log data" and analyzing human behavior, as *DDR Holdings*, 773 F.3d at 1258, cautions, "not all claims purporting to address Internet-centric challenges are eligible for patent." *DDR Holdings*, 773 F.3d at 1258, explains that the patent cannot "broadly and generically claim 'use of the Internet' to perform an abstract business practice (with insignificant added activity)."

1. Claims 1–11

Claim 1 of the '500 patent, the patent's "representative" method, states:

> 1. A method of detecting improper access of a patient's protected health information (PHI) in a computer environment, the method comprising:
>
> generating a rule for monitoring audit log data representing at least one of [the] transactions or activities that are executed in the computer environment, which are associated with the patient's PHI, the rule comprising at least one criterion related to accesses in excess of a specific volume, accesses during a pre-determined time interval, accesses by a specific user, that is indicative of improper access of the patient's PHI by an authorized user wherein the improper access is an indication of potential snooping or identity theft of the patient's PHI, the authorized user having a pre-defined role comprising authorized computer access to the patient's PHI;
>
> applying the rule to the audit log data to determine if an event has occurred, the event occurring if the at least one criterion has been met;
>
> storing, in a memory, a hit if the event has occurred; and
>
> providing notification if the event has occurred.

'500 patent, col. 16, ll. 27–46.

In other words, Claim 1 comprises (1) generating a rule "related to" the number of accesses, the timing of accesses, and the specific users in order to review "transactions or activities that are executed in a computer environment"; (2) applying the rule; (3) storing the result; and (4) announcing the result. None of these steps necessarily requires the use of a computer or any other technology. Rather, a person using "the human mind, or . . . using a pen and paper," *CyberSource Corp.*, 654 F.3d at 1372, can generate a rule for reviewing "audit log data" (i.e., a record of

activity) based on specific criteria, can apply the rule, can record the result, and can announce the result. Because the human mind can perform each step, Claim 1's method is unpatentable. *CyberSource Corp.*, 654 F.3d at 1373 ("[C]omputational methods which can be performed entirely in the human mind are the types of methods that embody the basic tools of scientific and technological work that are free to all men and reserved exclusively to none." (internal quotation marks omitted)); *see also Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat. Ass'n*, 776 F.3d 1343, 1347 (Fed. Cir. 2014) (invalidating patents that claimed a method of extracting data from documents, recognizing specific information, and storing that information in a memory because "the concept of data collection, recognition, and storage is undisputedly well-known" and "humans have always performed these functions").

None of the steps in Claim 1's method transforms the abstract idea into a patentable concept. Although the first step of the method requires "generating a rule for monitoring audit log data," Claim 1 neither states a rule nor instructs a computer to generate a rule. Instead, in at least one embodiment of the invention, "the rule is created by the user and/or a third party, such as a consultant with particular knowledge as to fraud or misuse of the particular type of data." '500 patent, col. 13, ll. 11–13. Also, the function performed by the computer in each remaining step of Claim 1's method is "purely conventional." *Alice*, 134 S. Ct. at 2358. Using a computer to apply a rule is elemental computing — the most basic

function of a computer. Similarly, using a computer to record a result and to announce a result are "well-understood, routine, conventional activities previously known to the industry." *Alice*, 134 S. Ct. at 2359 (internal quotation marks omitted).

Even considered as "an ordered combination," the steps of Claim 1's method add "nothing significantly more than an instruction to apply the abstract idea . . . using some unspecified, generic computer." *Alice*, 134 S. Ct. at 2360. In other words, "[t]his ordered combination of steps recites an abstraction — an idea, having no particular concrete or tangible form." *Ultramercial, Inc. v. Hulu, LLC*, 772 F.3d 709, 715 (Fed. Cir. 2014). Thus, the steps of the method are not "'enough' to transform an abstract idea into a patent-eligible invention." *Alice*, 134 S. Ct. at 2360.

Further, none of Claim 1's dependent claims adds a meaningful limitation to bring the abstract idea within the scope of Section 101. For example, Claim 2 adds "normalizing" or formatting the data; Claim 3 adds obtaining an authorized user's "role information"; and Claims 4, 5, and 6 add tracking an authorized user's access, volume of access, and time of access.

Finally, the abstract idea remains unpatentable despite the patent's effort to limit the invention to one field (health information) and to one technology (a computer). *See Bilski v. Kappos*, 561 U.S. 593, 612 (2010) ("[L]imiting an abstract idea to one field of use or adding token postsolution components d[oes] not make [a] concept patentable."); *Accenture Global Servs., GmbH v. Guidewire Software, Inc.*, 728 F.3d 1336, 1345 (Fed. Cir. 2013) (invalidating under Section 101 a patent despite

the patent's "attempt[] to limit the abstract concept to a computer implementation and to a specific industry").

2. Claims 12 and 13

Claim 12, which describes a system that implements on a generic computer Claim 1's method, is not patentable. Like the system in *Alice*, the system in Claim 12 contains a "handful of generic components." Specifically, Claim 12's system comprises an "interface" and a "microprocessor," both of which are fundamental components of every computer. "As a result, none of the hardware recited by the system claims offers a meaningful limitation beyond generally linking the use of the method to a particular technological environment, that is, implementation via computers." *Alice*, 134 S. Ct. at 2360 (internal quotation marks omitted).

Thus, Iatric correctly argues that "[t]he patent tethers an abstract idea — analyzing records to detect suspicious behavior — to a general purpose computer, a classic example of patent ineligibility under Section 101." (Doc. 50 at 11) Because the system adds no meaningful limitation to the method, Claim 12 is unpatentable for the same reasons as Claim 1. Similarly, Claim 13, a dependent claim that contains the additional limitation of "tracking access by the authorized user," is not an inventive concept that renders Claim 12 patentable.

3. Claims 14–17

Claim 14, which describes a computer-readable medium that contains instructions to perform Claim 1's method, fails for the same reasons. The patent states

that the computer-readable medium "can be any available media which can be accessed by a general purpose or special purpose computer," such as "RAM, ROM, EPROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices." '500 patent, col. 15, ll. 30–36. The patent's invocation of generic computer-readable media to perform the method adds no inventive concept to the underlying abstract idea. *See CyberSource Corp.*, 654 F.3d at 1375 (finding that the use of a computer-readable medium to verify credit card transactions and to detect fraud is an unpatentable abstract idea). None of Claim 14's dependent claims compels a different result.

CONCLUSION

Iatric's motion (Doc. 50) for oral argument is **DENIED**. Iatric's motion (Doc. 50) to dismiss is **GRANTED**, and the complaint is **DISMISSED WITHOUT PREJUDICE**. Under Section 101, the '500 patent is invalid. No later than **JULY 9, 2015**, FairWarning may amend the complaint to assert a claim that is independent of the '500 patent's validity. If FairWarning fails to amend the complaint on or before July 9, 2015, an order will promptly dismiss this action with prejudice.

ORDERED in Tampa, Florida, on June 24, 2015.

STEVEN D. MERRYDAY
UNITED STATES DISTRICT JUDGE

UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA
TAMPA DIVISION

FAIRWARNING IP, LLC

Plaintiff,

v.

Case No. 8:14-cv-02685-SDM-MAP

IATRIC SYSTEMS, INC.

Defendant.
_____/

IATRIC SYSTEMS, INC.'S DISPOSITIVE MOTION TO DISMISS AMENDED COMPLAINT UNDER 35 U.S.C. § 101 AND REQUEST FOR ORAL ARGUMENT

Pursuant to Rule 12(b)(6) of the Federal Rules of Civil Procedure, and Local Rule 3.01, Iatric Systems, Inc. respectfully moves this Court on the following grounds for an order dismissing FairWarning, IP, LLC's Amended Complaint:

(1) The patent asserted by FairWarning, U.S. Patent No. 8,578,500 ("the '500 Patent"), claims an abstract idea and is thus invalid under 35 U.S.C. § 101.

(2) Patentability under Section 101 of the Patent Act is a threshold legal issue, which should be addressed at the onset of litigation.

Thus, the Court should dismiss FairWarning's Amended Complaint under Rule 12(b)(6).

WHEREFORE, defendant, Iatric Systems, respectfully requests that the Court:

a. Grant this motion;

b. Enter an order declaring all of the claims of FairWarning's patent invalid and dismissing its Amended Complaint with prejudice; and,

c. Grant all such further relief as this Court deems just and appropriate.

MEMORANDUM IN SUPPORT OF MOTION TO DISMISS

In accordance with Local Rule 3.01, Iatric submits this memorandum of law in support of its motion.

I. INTRODUCTION

Although Iatric is the pioneer and prior entrant in the field of patient privacy monitoring software, FairWarning has accused Iatric of infringing its '500 Patent. FairWarning's Amended Complaint must be dismissed, however, because the '500 Patent is invalid under Section 101 of the Patent Act. The Supreme Court has made clear that patents may not claim abstract ideas. FairWarning's patent claims the abstract idea of analyzing records of human activity to detect suspicious behavior, and its direction to "apply it" in the computer context fails to describe an improvement to the function of a computer itself, or an improvement in another technological field. Pre-existing federal HIPAA requirements demonstrate that applying this idea to health records, computerized or otherwise, does not provide the "inventive step" required by Supreme Court precedent. As such, FairWarning's patent is invalid under 35 U.S.C. § 101.

II. STATEMENT OF FACTS

A. FairWarning's '500 Patent and "Fraud and Misuse Detection"

In its Amended Complaint, FairWarning alleges Iatric's software infringes all claims of the '500 Patent. Amended Complaint at ¶¶ 10, 13, 17, 29 and 38. The claims of the '500

Patent are directed to the concept of analyzing records of human activity to detect suspicious behavior. The '500 Patent employs this concept to detect improper access to patient medical records, and is entitled "System and Method of Fraud and Misuse Detection."

The idea of analyzing records of human activity to detect suspicious behavior dates back thousands of years. In ancient Egypt, the Pharaoh had scribes account for his or her gold and other assets; these scribes were charged with fraud prevention and detection. More modern examples include the government's use of an IRS audit to convict Al Capone of tax evasion in the 1930s,[1] the use of evidence logs to detect tampering with evidence,[2] the use of internal auditing to expose over $3 billion in financial fraud at WorldCom in 2002,[3] and federal fraud investigators auditing hospital patient records to detect Medicare/Medicaid billing fraud.[4]

The records of human activity in this instance are audit log data (that is, records of computer use by a computer user), and the inappropriate human activity is improper access to protected health information. The '500 Patent, attached as Exhibit A, states:

> The invention relates to a system and method of detecting fraud and/or misuse in a computer environment based on analyzing data such as in log files, or other similar records, including user identifier data. More particularly, the invention relates to a

[1] *U.S. v. Capone*, 93 F.2d 840 (7th Cir. 1937), *cert. denied*, 58 S.Ct. 750 (1938); *Gibson v. Maryland*, 771 A.2d 536, 543 (Md. App. 2001) (noting that the IRS audited Al Capone's income tax returns while the government was on his trail for bootlegging).

[2] *See, e.g., Haley v. Mississippi*, 737 So.2d 371, 375 (Miss. App. Ct. 1998) (defendant argued that errors on evidence log suggested tampering with the evidence).

[3] *In re WorldCom, Inc. Securities Litigation*, 388 F. Supp. 2d 319, 331 (S.D.N.Y. 2005) (WorldCom's Internal Audit department uncovered WorldCom's financial fraud through internal audits performed beginning in early 2002; the fraud was disclosed to the public in June 2002).

[4] *See, e.g., University of Medicine & Dentistry of New Jersey v. Corrigan*, 347 F.3d 57, 64-67 (3d Cir. 2003).

system and method of detecting fraud and/or misuse in a computer environment based on analyzing application layer data such as in log files, including user identifier data. (Col 1, lines 15-21.)

The patent contains three independent claims (claims 1, 12, and 14), and fourteen dependent claims (claims 2-11, 13, and 15-17).

1. Independent Claims (Claims 1, 12, and 14)

Claim 1 of the '500 Patent is a method claim that reads:

1. A method of detecting improper access of a patient's protected health information (PHI) in a computer environment, the method comprising:

   generating a rule for monitoring audit log data representing at least one of transactions or activities that are executed in the computer environment, which are associated with the patient's PHI, the rule comprising at least one criterion related to accesses in excess of a specific volume, accesses during a pre-determined time interval, accesses by a specific user, that is indicative of improper access of the patient's PHI by an authorized user wherein the improper access is an indication of potential snooping or identity theft of the patient's PHI, the authorized user having a pre-defined role comprising authorized computer access to the patient's PHI;

   applying the rule to the audit log data to determine if an event has occurred, the event occurring if the at least one criterion has been met;

   storing, in a memory, a hit if the event has occurred; and

   providing notification if the event has occurred.

In plain English, claim 1 covers the simple, abstract idea of analyzing records of human behavior (monitoring audit log data), determining if accesses to patient medical records are (a) in excess of a specified volume, (b) during a specified time of day, and/or (c) made by a specific user (detecting suspicious behavior), and if suspicious behavior is detected, storing that fact in a memory and providing notification.

Claim 12 is similar to claim 1, but has been adapted to claim a system rather than a method. Claim 12 is directed to a "system for detecting improper access of a patient's protected health information (PHI) in a health-care system computer environment," with the system comprising a "user interface" and a "microprocessor" that implement the method of claim 1. As such, the system of claim 12 analyzes records of human activity (audit log data) to detect suspicious activity (improper access).

Claim 14 is also similar to claim 1, but claim 14 has been drafted to claim a "computer-readable medium" for performing the method of claim 1 rather than the method itself. Claim 14 is directed to a "non-transitory computer-readable medium" having software "for performing a method of detecting improper access of a patient's protected health information (PHI) in a health-care system computing environment." As such, the computer-executable instructions of claim 14 analyze records of human activity (called "transactions and activities" in claim 14) to detect suspicious activity (improper access).

2. Dependent Claims (Claims 2-11, 13, 15-17)

The dependent claims in the '500 Patent do not alter the abstract nature of what is claimed. They merely present insignificant variations on the method of monitoring suspicious activity. Claim 2 adds the routine, conventional step of formatting the data to be used in the analysis. Claims 3-4, 13, and 16 describe variations on the rules or intended use of the rules applied to detect improper access. Claim 3 describes a rule based on an authorized user's role. Claims 4, 13, and 16 describe criteria for determining misuse by tracking access to the information of another person.

Claims 5-11, 15, and 17 similarly describe various criteria that can be applied. This includes: the access tracked comprises access in excess of a specific volume, which is a criterion in claim 1 (claim 5); the access tracked comprises access over a predetermined time interval, which is also a criterion in claim 1 (claim 6); using a relation between the authorized user and another person to detect various types of snooping (claim 7); analyzing data of a selected patient or user (claim 8); analyzing whether the information was from someone discharged in the past (claim 9); analyzing the relationship between the user and the patient (claim 10); analyzing whether the access occurred outside the user's normal work hours (claim 11). Claim 15 describes using criterion that is indicative of fraudulent claims. Claim 17 describes that the access tracked includes access to a predetermined amount of the information of another person.

B. Relevant HIPAA Regulations

The Health Insurance Portability and Accountability Act of 1996 ("HIPAA") fostered regulations protecting the privacy and security of certain health information. The measures that healthcare providers must take to safeguard patient information essentially embody the concept to which the '500 Patent is directed:

> A covered entity ... must . . . [i]mplement ***hardware, software***, and/or procedural mechanisms ***that record and examine activity in*** information systems that contain or use electronic ***protected health information***.

45 C.F.R. §164.312(b) (emphasis added). Similarly:

> A covered entity ... must . . . [i]mplement policies and procedures to prevent, detect, contain, and correct security violations ... [which includes] [i]mplement[ing] procedures to ***regularly review records of information system activity, such as audit logs***, access reports, and security incident tracking reports.

45 C.F.R. §164.308(a)(1), at (i) and (ii)(D) (emphasis added). The HIPAA regulations also suggest that the security violations of interest could include unauthorized use by authorized users. 45 C.F.R. § 164.308(a)(1)(ii)(C) requires appropriate sanctions for those workforce members (*i.e.*, authorized users[5]) who fail to comply with security policies and procedures (*i.e.*, misuse or unauthorized access). All of these sections were published at 68 Fed. Reg. 8376 on February 20, 2003, more than two years before Mr. Long filed the provisional patent application to which the '500 Patent claims priority. Ex. A, '500 Patent, at Col. 1, lines 7-11. FairWarning's patent claims do little more than mirror these regulations, describing generic computer tasks by which activity in information systems that contain protected health information is examined and reported.

III. ARGUMENT

A. Iatric's Motion to Dismiss Is a Proper Vehicle for Determining the Patent Eligibility of FairWarning's Patent

Patentability under 35 U.S.C. §101 is a threshold legal issue, and its proper resolution at the onset of litigation will conserve judicial and party resources. The issue of patentable subject matter is purely an issue of law. *See In re Bilski*, 545 F.3d 943, 951 (Fed. Cir. 2008) (*en banc*), *aff'd sub nom. Bilski v. Kappos*, 561 U.S. 593 (2010).

As Judge Mayer of the Federal Circuit explains:

> [T]he section 101 determination bears some of the hallmarks of a jurisdictional inquiry in that a court must . . . first assess whether claimed subject matter is even *eligible* for patent protection before addressing questions of invalidity or infringement ... Failure to recite statutory subject matter is the sort of basic deficiency that can, and should, be exposed at the point of minimum expenditure of time and money by the parties and the court... [R]esolving subject matter eligibility at the outset provides a

[5] 45 C.F.R. § 164.308(3)(i) requires access to electronic PHI for all workforce members.

> bulwark against vexatious infringement suits. The scourge of meritless infringement claims has continued unabated for decades due, in no small measure, to the ease of asserting such claims and the enormous sums required to defend against them.

Ultramercial, Inc. v. Hulu, LLC, 772 F.3d 709, 718-19 (Fed. Cir. 2014) (Mayer, J., concurring) (internal quotes omitted). Similarly:

> From a practical perspective, there are clear advantages to addressing Section 101's requirements at the outset of litigation. Patent eligibility issues often can be resolved without lengthy claim construction, and an early determination that the subject matter of asserted claims is patent ineligible can spare both litigants and courts years of needless litigation.

I/P Engine, Inc. v. AOL Inc., 576 F. App'x 982, 995-96 (Fed. Cir. 2014) (Mayer, J., concurring).

As noted by Judge Mayer, claim construction is not a prerequisite to determining patent eligibility. *Ultramercial*, 722 F.3d at 1339 (no rule requiring district courts to construe claims before determining subject matter eligibility); *Bancorp Svcs., L.L.C. v. Sun Life Assurance Co. of Canada (U.S.)*, 687 F.3d 1266, 1273 (Fed. Cir. 2012) (claim construction is not a prerequisite to a validity determination under § 101); *see, e.g.*, *Bilski v. Kappos*, 561 U.S. 593 (2010) (finding subject matter ineligible for patent protection without claim construction). Here, there are no disputed claim terms that need to be construed for the Section 101 analysis. If a patentee were able to defeat a motion to dismiss based on Section 101 by raising the mere possibility of a future claim construction dispute, the salutary benefit of such challenges will rarely be realized. Thus, unlike in other cases, there is no need to defer determination of the Section 101 issue until after claim construction. *StoneEagle Services, Inc. v. Pay-Plus Solutions, Inc.*, 2015 WL 518852, at *4 (M.D. Fla. Feb. 9, 2015)

(Hernandez Covington, J.) (both parties disputed the construction of the claim term "stored-value card").

Moreover, while courts primarily consider the allegations in the complaint in determining whether to grant a Rule 12(b)(6) motion, they may also take matters of public record into account, including court cases, statutes, regulations, and U.S. patent documents. *Bryant v. Avado Brands, Inc.*, 187 F.3d 1271, 1280 (11th Cir. 1999); *Spasojevic v. Wells Fargo Bank, N.A.*, 2014 WL 222942, at *1 (M.D. Fla. Jan. 21, 2014) (Merryday, J.); *Pani v. Empire Blue Cross Blue Shield*, 152 F.3d 67, 75 (2d Cir. 1998) (case law and statutes properly considered on motion to dismiss); *Sebastian v. U.S.*, 185 F.3d 1368, 1374 (Fed. Cir. 1999) (historical regulations were matters of public record properly considered on a motion to dismiss); *Goldthread v. Davison*, 2007 WL 2471803, at *4, n.2 (M.D. Tenn. Aug. 29, 2007) (granting motion to dismiss and considering patent documents as matters of public record).[6] Iatric does not rely on materials outside of the pleadings and the public record and thus its motion is ripe for decision. *Contrast StoneEagle*, 2015 WL 518852, at *4 & n.2 (plaintiff referred to expert report and declaration from plaintiff's chairman and CEO, materials not properly considered on a motion to dismiss; defendants also relied on matters outside the pleadings, *but provided no authority* that would allow the material to be considered on a Rule 12 motion).

Indeed, numerous courts have evaluated patents for abstractness at the pleadings stage. *See, e.g., Content Extraction & Transmission LLC v. Wells Fargo Bank, N.A.*, 776

[6] *See also Snap-On Inc. v. Hunter Engineering Co.*, 29 F. Supp. 2d 965, 969 (E.D. Wis. 1998) (records from the USPTO on a motion to dismiss considered "because they are public records"); *Coinstar, Inc. v. Coinbank Automated Systems, Inc.*, 998 F. Supp. 1109, 1114 (N.D. Cal. 1998) (appropriate judicial notice of matters of public record includes patents and patent file histories).

F.3d 1343, 1349 (Fed. Cir. 2014) (affirming grant of motion to dismiss); *buySAFE, Inc. v. Google, Inc.*, 765 F.3d 1350, 1351-52 (Fed. Cir. 2014) (affirming judgment on the pleadings); *Ultramercial,* 772 F.3d at 717 (affirming grant of motion to dismiss); *Open Text S.A. v. Alfresco Software Ltd.*, 2014 WL 4684429, at *5 (N.D. Cal. Sept. 19, 2015) (granting motion to dismiss under Section 101).

B. FairWarning's Patent Fails the Supreme Court's Test for Patent-Eligible Subject Matter

Section 101 of the Patent Act defines the subject matter that is eligible for patent protection.[7] The Supreme Court has imposed three important limits, ruling that laws of nature, natural phenomena, and abstract ideas do not fall within the subject matter that can be protected by the patent laws. *Bilski*, 561 U.S. at 601; *Mayo Collaborative Svcs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1293 (2012); *Alice*, 134 S. Ct. at 2354.

In its decision in the *Alice* case – which issued after FairWarning's '500 Patent was issued – the Supreme Court clarified the two-step framework for determining patent eligibility under 35 U.S.C. § 101 – and its application to computer-related claims. *See Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347 (2014). *Alice* held that methods of organizing human activity, like those in the '500 Patent, are abstract ideas and that application of generic technologies such as general purpose computers cannot render such abstract ideas patentable. 134 S. Ct. at 2355–56, 2358. The Supreme Court's decision in *Alice* makes clear that merely tying an abstract idea to a computer is not sufficient to transform that idea into a patent-eligible invention.

[7] "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." 35 U.S.C. § 101.

This Court has observed correctly that the judicial tide has turned decisively against the patent eligibility of software patents. *See Every Penny Counts, Inc. v. Wells Fargo Bank N.A.*, 2014 WL 4540319, at *4 n.4 (M.D. Fla. Sept. 11, 2014) (Merryday, J.) ("Conspicuously, the Supreme Court vacated the only Federal Circuit opinion, *Ultramercial*, upholding a software patent and declined certiorari over the two actions, *Bancorp* and *Accenture*, that invalidate software patents."). Indeed, following the Supreme Court's guidance, although it is hard to give a one-size-fits-all definition, the Federal Circuit has had little difficulty identifying abstract ideas, including:

- "credit card fraud detection," *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1373 (Fed. Cir. 2011);
- "generating tasks [in an insurance organization] based on rules to be completed upon the occurrence of an event," *Accenture Global Servs. v. Guidewire Software, Inc.*, 728 F.3d 1336, 1344 (Fed. Cir. 2013) (alteration omitted), *cert. denied*, 134 S. Ct. 2871 (2014); and
- "categorical data storage," *Cyberfone Sys., LLC v. CNN Interactive Grp., Inc.*, 558 F. App'x 988, 992 (Fed. Cir. 2014).

FairWarning's patent takes the abstract idea of analyzing records of human activity to detect suspicious behavior and implements it on computer to monitor access to medical records. The patent tethers an abstract idea—analyzing records to detect suspicious behavior—to a general purpose computer, a classic example of patent ineligibility under Section 101. Indeed, at least one other court has already found a patent invalid on similar grounds. S*ee MyMedicalRecords, Inc. v. Walgreen Co.*, 2014 WL 7339201, at *3 (C.D. Cal. Dec. 23, 2014) (patent claiming method of collecting, accessing, and managing personal health records in a secure and private manner held patent-ineligible under Section 101).

1. First Step: FairWarning's Claims Are Directed to an Abstract Idea

Step one described by the Supreme Court in *Alice* asks whether the claims are directed to an abstract idea (or a law of nature or natural phenomenon). The analysis must be focused on the *claims as written*. *Bascom Research, LLC v. LinkedIn, Inc.*, --- F. Supp. 3d ---, 2015 WL 149480, at *6 (N.D. Cal. Jan. 5, 2015) (at step one of the *Mayo/Alice* test, a court must evaluate the patent claims on their face and determine if the claims are drawn to a patent ineligible concept such as an abstract idea); *Alice*, 134 S.Ct. at 2356 (evaluating claims "[o]n their face" for step one). The Court in *Alice* found the patent at issue invalid under Section 101 because the concept of "intermediated settlement" was an abstract idea, like the concept of hedging risks that the Court had earlier addressed in *Bilski*, 561 U.S. 593. *Alice*, 134 S. Ct. at 2356. The Court held the patent invalid *even though the patent claims included a number of specific steps*, such as creating and adjusting shadow credit records for stakeholders. *Id.* at 2352 n.2. Here, the claims of FairWarning's patent also recite a set of steps, but this likewise does not obscure the fact that the patent claims the abstract idea of analyzing records of activity to detect inappropriate conduct. *See* Exhibit B (chart comparing patent claims held patent ineligible in *Alice* and *Ultramercial* cases to claim 1 of the '500 Patent).

In evaluating the abstract nature of a patent's claims, the correct approach, set forth by the Supreme Court in *Bilski, Mayo* and *Alice*, is to first determine the central idea addressed by the claims. The Supreme Court has repeatedly confirmed that abstract ideas should be characterized at a relatively high level of generality and has rejected efforts to frame ideas more narrowly in light of specific claim limitations. In *Bilski*, the Supreme

Court identified the idea to which the claims were directed as simply "hedging, or protecting against risk." 561 U.S. at 611. The Court used this characterization even though the independent claims listed a series of specific steps (including a mathematical formula in one) and the asserted dependent claims further limited the context to energy markets and applied specified statistical techniques. *Id.* at 599-600.

Likewise, in *Alice*, the claimed scheme for mitigating "settlement risk" involved using a computer system as a third-party intermediary to create "shadow" credit and debit records (*i.e.*, account ledgers) to mirror balances in the parties' real-world accounts at banks. The intermediary updates the shadow records as transactions are entered, allowing "only those transactions for which the parties' updated shadow records indicate sufficient resources to satisfy their mutual obligations." *Alice*, 134 S.Ct. at 2352. Distilling these details of the claims, the Court found that the patent claims were directed to the basic and abstract idea of "intermediated settlement." *Id.* at 2355.

These cases confirm an important point: determining the idea to which a claim is directed does not entail a searching inquiry into claim limitations. *See, e.g., Open Text S.A. v. Box, Inc.*, 2015 WL 269036, at *3 (N.D. Cal. Jan. 20, 2015). In *Open Text v. Box*, the plaintiff argued that the idea behind the claim (which it denied was abstract) was a "system for providing a collaborative workspace on a network server." *Id.* However, the court rejected plaintiff's attempt to "smuggle[] in" a reference to a network server from the claim limitations. *Id.* The court emphasized that the idea behind the claim must be "[s]horn of its implementation-specific fleece." *Id.* The court found the claim at issue was directed at the abstract idea of providing a method for people to collaborate and share information without

the need for specialized software or expertise. *Id.* Similarly, the Court here should reject FairWarning's requests to smuggle claim limitations such as audit logs into the idea behind the claims.

Using claim limitations to define the idea to which a claim is directed would conflate step one with step two, contrary to the Supreme Court's approach. *See id.* (noting that any novelty in implementation of the idea is a factor to be considered only in the second step of the *Alice* analysis); *Ultramercial*, 772 F.3d at 715 (same). This Court followed the same exercise in *Every Penny Counts* by looking past several claim limitations and characterizing the asserted patents as drawn to the basic concept of "salami slicing," that is, regularly capturing tiny amounts and retaining them until they accumulate into a large amount. 2014 WL 4540319, at *4 & n.5. The Court deemed this an abstract idea. *Id.* FairWarning's patent is likewise accurately viewed as directed to the abstract idea of analyzing records of activity for indications of improper conduct.

Similarly, the fact that an abstract idea can be described in different ways is not evidence or proof that the idea is not abstract. *See Tenon & Groove, LLC v. Plusgrade S.E.C.*, 2015 WL 1133213, at *3 (D. Del. Mar. 11, 2015) ("the court fails to understand how the use of slightly different words to describe something *abstract* is proof that it is not abstract"). The English language is capable of conveying like ideas in different terms. *Id.*; *see also Oracle America, Inc. v. Google Inc.*, 750 F.3d 1339, 1367 (Fed. Cir. 2014) (original work entitled to copyright protection so long as the author had "multiple ways to express the underlying idea"). Courts have rejected the argument that uniform phrasing must be used to describe abstract ideas. *See, e.g., Tenon & Groove*, 2015 WL 1133213, at *3.

i. FairWarning claims the kind of abstract idea that numerous opinions have found patent-ineligible

Since *Alice*, the Federal Circuit and district courts have repeatedly invalidated computerized data-processing patents that are similar to FairWarning's asserted claims.[8] For example, in *MyMedicalRecords*, the patent claimed a method of collecting, accessing, and managing personal health records in a secure and private manner, noting several purported shortcomings in the prior art. 2014 WL 7339201, at *3. The district court found that the claims of the patent were impermissibly abstract because the "concept of secure record access and management, in the context of personal health records or not, is an age-old idea." *Id.* at *3. The court then granted a motion for judgment on the pleadings based on patent ineligibility under 35 U.S.C. § 101. *Id.*

The Federal Circuit's decision in *DDR Holdings* supports dismissal in this case as it demonstrates the stark contrast between the claims the court found valid and the claims of the '500 Patent at issue here. In *DDR Holdings*, the claims described a system that would generate a hybrid web page combining the host website and the third-party merchant website. *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1257 (Fed. Cir. 2014) (software claims were patent eligible where the "claimed solution [was] necessarily rooted in computer technology" to overcome a problem specifically arising in the realm of computer networks). The Federal Circuit emphasized that the claims in *DDR Holdings* "stand apart" because they

[8] *See, e.g.*, *Comcast IP Holdings I, LLC v. Sprint Comms. Co.*, 2014 WL 3542055, at *5 (D. Del. July 16, 2014) (invalidating a claim of the patent at issue because it "merely covers the application of what has for a long time been conducted solely in the mind to modern, computerized, telephony networks"); *Walker Digital, LLC v. Google, Inc.*, 2014 WL 4365245, at *6 (D. Del. Sept. 3, 2014) ("Even accepting that the use of a computer increases speed and efficiency of performing the steps in the claims, and improves the likelihood of preserving the anonymity of the first and second parties, these characteristics do not save the claims").

"specified how interactions with the internet are manipulated to yield a desired result" in which the internet did not operate "in its normal, expected manner," thus describing a *technological solution to a technological problem*, and do not merely recite the performance of some conventional business practice on the internet, a commonplace business method aimed at processing business information, or applying a known business process to a "particular technological environment." *Id.* at 1257, 1259.

Here, unlike *DDR Holdings*, not one of the claims of the '500 Patent describes a technological solution that causes a computer to operate outside its normal, expected manner. Instead, the claims describe nothing more than a modern spin on the millennia-old method for examining human activity for evidence of wrongdoing. Whether using a computer would increase speed or efficiency in undertaking that task is irrelevant.[9] The fact that the '500 Patent applies this known concept to the "particular technological environment" of electronic personal health records is not enough to save it. *See DDR Holdings* at 1259.

FairWarning may harp on the implementation of its claimed method on computers (*e.g.*, references to audit logs or normalizing audit logs), but the *claims themselves* do not describe anything other than the routine, conventional idea of reviewing audit logs. The claims themselves do not specify any particular technological mechanisms for creating or normalizing audit logs. The important inquiry for a Section 101 analysis is to *look to the claims*, not the specification. *Accenture*, 728 F.3d at 1345 (claims were patent ineligible under Section 101 despite "detailed software implementation guidelines" in the

[9] *Walker Digital*, 2014 WL 4365245, at *6 ("Even accepting that the use of a computer increases speed and efficiency of performing the steps in the claims ... these characteristics do not save the claims").

specification). FairWarning's emphasis on unclaimed features is irrelevant, nothing more than a red herring. The '500 Patent claims merely describe the abstract idea of examining human activity for evidence of wrongdoing by using a generic computer and conventional and generic computer technology (*i.e.*, audit logs). Therefore, the "claimed solution" is *not* "rooted in computer technology." *See MyMedicalRecords* at *4-5 (distinguishing *DDR Holdings* on the basis that the '466 Patent recited an invention that was "merely the routine and conventional use of the Internet and computer[s]" for "collecting, accessing, and managing health records in a secure and private manner."); *Open Text v. Box*, 2015 WL 269036, at *5 (distinguishing *DDR Holdings* because there was no indication that the claimed "web browser" was used in any non-routine and unconventional way).

ii. FairWarning's patent on an abstract idea raises the specter of preemption

Several courts, including the Supreme Court, have cited concern over preemption to further explain the need for an "abstract idea" exception. In *Bilski,* the Supreme Court worried that "[a]llowing petitioners to patent risk hedging would preempt use of this approach in all fields, and would effectively grant a monopoly over an abstract idea." 130 S.Ct. at 3231. The same risk is apparent here.

Federal regulations under HIPAA require "covered entities" to maintain the security of patient information. FairWarning's patent claims do little more than mirror these regulations, describing generic computer tasks by which records of user activity in information systems that contain protected health information (*i.e.*, audit logs) are examined and reported through a generic database query. Patenting compliance with HIPAA regulations obviously threatens to pre-empt the field. Similarly, the court in

MyMedicalRecords found that since the concept of "secure record access and management, in the context of personal health records or not, is an age-old idea" a patent on this concept would threaten to preempt the entire field of secure filing systems. 2014 WL 7339201, at *4-5.

Additionally, while the "pre-emption concern" may "undergird" the Supreme Court's Section 101 jurisprudence, *Alice*, 134 S.Ct. at 2358, it does not follow that courts determine patentability by guessing at the probability of pre-emption. *Open Text v. Box*, 2015 WL 269036, at *4. There is no non-speculative way for a court to determine whether and to what extent future innovation might be curtailed. *Id.* Thus, the preemption concern is largely already "baked into" the *Alice* test. *Id.* Because of this, even "fairly specific requirements" in claims cannot save them. *Id.*; *buySAFE*, 765 F.3d at 1353 ("exclusion applies if a claim involves ... [an] abstract idea, even if the particular ... abstract idea at issue is narrow"). Just as specific "activity log" requirements could not save the claims in *Ultramercial*, so too the "audit log" requirements cannot save the claims here. *Ultramercial*, 772 F.3d at 712.

2. Second Step: FairWarning's Claims Do Not Contain Additional Features Sufficient to Transform the Abstract Idea

Step two in the analysis laid out by the Supreme Court asks whether the claims contain "additional features," specifically, an "inventive concept sufficient to transform the claimed abstract idea into a patent-eligible application." *Alice*, 134 S.Ct. at 2357 (quoting *Mayo*) (internal quotes omitted). A court may consider the elements of the asserted claims both individually and as an "ordered combination." *Id.* at 2355. Taking an abstract idea and simply breaking it up into a series of steps or adding "well-understood, routine, conventional activities previously known to the industry" contributes nothing inventive. *Id.* at 2359

(quoting *Mayo*) (internal quotes omitted). In particular, "[t]he introduction of a computer into the claims does not alter the analysis at *Mayo* step two." *Id.* at 2357.

Although Alice Corp.'s claims purported to describe "a computerized scheme for mitigating 'settlement risk'" broken into several discrete steps, the method claims ultimately "simply recite[d] the concept of intermediated settlement as performed by a generic computer." *Id.* at 2352, 2359. They did not, "for example, purport to improve the functioning of the computer itself," or "effect an improvement in any other technology or technical field." *Id.* at 2359. In short, the method claims, like the system and readable medium claims, added nothing of substance to the underlying abstract idea. *Id.* at 2360. "[G]eneric computer components" do not become patent eligible simply upon being "configured" to perform "specific computerized functions." *Id.*

As in *Alice*, FairWarning's claims add no meaningful limitations to convert the abstract concept into a patent-eligible application.[10] The issue for Section 101 analysis is whether the individual process steps are merely well-understood, routine, conventional activities previously known to the industry. And as the HIPAA regulations establish, reviewing audit logs to detect potential fraud or misuse of patient health information, including by authorized users, was nothing more than conventional (and indeed required) activity in the industry well before the priority date for the '500 Patent. *See* above at p. 6-7. Any contrary assertion by FairWarning is simply not plausible.

[10] The required technological limitations must appear in the patent *claims* themselves. Like other requirements for patentability, Section 101 looks to individual claims, not a patent as a whole. *Alice*, 134 S. Ct. at 2355. It is generally irrelevant whether a patent's *specification* describes inventive, technological limitations. The question is whether each claim contains such limitations and thereby restricts itself to patent-eligible subject matter. *See Accenture*, 728 F.3d at 1345.

The few steps of both Fair Warning's method and system claims "simply instruct the practitioner to implement the abstract idea of" analyzing records of human activity to discern indications of improper purpose "on a generic computer." *Alice*, 134 S. Ct. at 2359. Indeed, the claims rejected by the Supreme Court in the *Alice* case provided much more detail about the function the computers performed, and yet that detail did not save the claims. *Id.* at 2352 and n.2, 2359 (shadow account ledgers, updated in real time, etc.). The table provided in Exhibit B shows FairWarning's claim language beside the patent-ineligible claim language from *Alice* and *Ultramercial*.

FairWarning's patent claims break the abstract idea into steps performed by the computer, all of which are, in *Alice*'s terminology, "purely conventional." *Alice*, at 2351. Claim 1, for example, calls for a computer to generate a rule for analyzing data, indubitably one of the most basic functions of a computer. Developing a search query to analyze data within a database or across multiple databases is familiar to all database users, such as lawyers searching in Lexis-Nexis, Westlaw, or PACER, which can involve multiple databases (*e.g.*, different court systems.). *See, e.g.*, *Ex Parte Morsa*, 2014 WL 986144, at *1, 3 (P.T.A.B. Feb. 25, 2014) (rejecting patent claims in IPR for application filed in 2001 where claims were "a classic example of a *conventional database query* in which a user submits a request to a database and receives stored information from the database responsive to the request") (emphasis added); PACER User Manual for ECF Courts, at 14-20, *available at* http://www.pacer.gov/documents/pacermanual.pdf.[11] For example, a lawyer may develop a

[11] Government documents available online, such as the PACER User Manual, are public records. *See Kitty Hawk Aircargo, Inc. v. Chao*, 418 F.3d 453, 457 (5th Cir. 2005) (taking judicial notice of approval published on federal agency's website); *Denius v. Dunlap*, 330 F.3d 919, 926-27 (7th Cir.

search query in Westlaw or PACER to find cases or decisions that meet the search query's criteria, such as cases issued or filed during a pre-determined time interval (like decisions issued after *Alice*), decisions written by a specific judge, or cases with specific parties.

The remaining steps of claim 1—applying the rule to the data to analyze that data, storing the results of that analysis (*e.g.*, a "hit") in computer memory if the event has occurred, and providing notification if a hit has occurred—are also purely conventional and would be familiar to users of typical computer databases. *See id.* Continuing with the Westlaw and PACER example, after developing a search query, the lawyer applies the query to the Westlaw or PACER databases, and Westlaw or PACER responds by generating a list of cases that match the query's criteria, and then notifies the lawyer by displaying that list on the lawyer's computer. *See id.* Under *Alice*, these are not "additional limitations" narrowing the abstract idea. Expressing an abstract concept as a "series of steps" does not make it patent eligible. *Id.* at 2356 (quoting *Bilski*).

Considering Claim 1 as "an ordered combination" does not help in revealing any "inventive concept." Instead, it exposes the claim as no more than a recital of the concept of detecting improper access to protected health information in a computer environment, or, in *Alice*'s words, "on a generic computer." *Id.* at 2359. As in *Alice*, when considered as an ordered combination, the computer components of FairWarning's method "ad[d] nothing ... that is not already present when the steps are considered separately." *See id.* at 2351.

2003) (taking judicial notice of information on official government website); *Paralyzed Veterans of America v. McPherson*, 2008 WL 4183981, at *5 (N.D. Cal. Sept. 9, 2008) (collecting other cases).

Claims 12 and 14, the "system" and "computer-readable medium" claims of the '500 Patent, mirror the method claims and therefore fail for the same reasons as in *Alice*. The system claim 12 merely applies the method of claim 1 using a generic "user interface" and "microprocessor." *See, e.g.*, *DietGoal Innovations LLC v. Bravo Media LLC*, 33 F. Supp. 3d 271, 288 (S.D.N.Y. 2014) (system claims were invalid under Section 101 where they merely described "generic computer components that can be found on any general-purpose computer, such as a user interface"); White House Office of Communications Fact Sheet on Export Controls on Computers, 2000 WL 121985, at *3 (Feb. 1, 2000) ("general purpose ... microprocessors ... are used in virtually all consumer and business personal computers"). Thus, the system claim (Claim 12) falls with the method claim (Claim 1).

The fate of claim 14 is no different. Simply directing claim 14 to a "computer-readable medium" for performing the method of claim 1 does not make claim 14 any more patent eligible than claim 1 itself. *See, e.g.*, *CyberSource*, 654 F.3d 1366 at 1373-76 (claim drawn to a "computer readable medium" was not patent eligible because it was drawn to the underlying method of credit card fraud detection, rejecting patentee's argument that it was patent eligible under the "machine-or-transformation" test). Similarly, *Alice* suggests that both the system and computer component claims of the '500 Patent should rise and fall with the method claims:

> None of the hardware recited by the system claims offers a meaningful limitation beyond generally linking the use of the [method] to a particular technological environment, that is, implementation via computers. ... Put another way, the system claims are no different from the method claims in substance. The method claims recite the abstract idea implemented on a generic computer; the system claims recite a handful of generic computer components configured to implement the same idea.

Alice, 134 S.Ct. at 2360 (internal quotations omitted); *DietGoal*, 33 F. Supp. 3d at 288-89. Because FairWarning's system and computer component claims "add nothing of substance to the underlying abstract idea," they also fail to claim patent-eligible subject matter required by § 101. *Id.*

The dependent claims do not change the analysis. The dependent claims merely add additional conventional steps or present variations on the details, criteria, or use of the rule:

- Claim 2 requires normalizing audit log data. But normalizing audit log data just organizes it into a desired format. There is nothing in the claims that describes any inventive step with respect to normalizing data. Normalizing data for computer processing was entirely conventional by the time of the '500 Patent's May 2005 priority date.[12]

- Claims 3 and 8 do not add anything different from the general ideas covered in the independent claims. Claim 3 describes the idea of obtaining role information and generating the rule based on the user's role. Claim 8 describes the idea of accessing select user data and applying the rule to it. Tailored searches such as these are entirely conventional use of generic database querying capability. *See, e.g.*, *Ex Parte Morsa*, 2014 WL 986144, at *3.

- Claims 7, 9-11, 15, and 17 merely describe other (common sense) criterion: relationships for family, VIP, or co-worker snooping (claim 7), patient discharged from medical facility but not recently (claim 9), user is remote and not a caregiver to patient (claim 10), access outside user's normal work hours (claim 11), criterion is indicative of fraudulent claims filed by user (claim 15), and access to a predetermined amount of information (claim 17). These are nothing more than common sense ideas about what constitutes suspicious activity.

- Claims 4, 13, and 16 all merely present a variation on the use of the rule: determining a misuse of the patient's PHI by tracking access by the authorized user of patient's PHI of another person. This is nothing more than a variation on the abstract idea of analyzing records of human activity to detect suspicious behavior.

[12] *See* Exhibit C, U.S. Patent No. 6,134,664 (issued Oct. 17, 2000), at 1 (title and abstract discuss "audit information ... [that] is normalized to a standard format" thereby reducing the computational requirements for "a misuse and intrusion detection engine"); Exhibit D, U.S. Patent Publication No. 2002/0174093 (published Nov. 21, 2002), at 3, ¶ 42 (for "audit logs produce[d] by ... different vendor [applications], ... scripts must extract the data from the audit logs" and this "extracted data must then be normalized").

- Claims 5 and 6 essentially restate criterion from claim 1: access in excess of a specific volume and access over a pre-determined time interval.

The fact that FairWarning's claims apply to detecting inappropriate activity in the context of protected health information using computers, rather than to databases in any number of other spheres, is a classic field-of-use limitation that cannot save FairWarning's claims. Such field of use restrictions do not make the abstract concept underlying FairWarning's claims any more patent-eligible than did limiting the hedging concept in *Bilski* to energy market transactions, 561 U.S. at 611, or the information clearinghouse concept in *Dealertrack* to car loan applications. *See Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1334 (Fed. Cir. 2012). The Federal Circuit has repeatedly emphasized that attempts to limit abstract ideas "to a particular technological environment" (such as the "audit logs" in a health-care system computer environment emphasized by Plaintiff) are "insufficient" to save claims under Section 101. *Content Extraction*, 776 F.3d 1343, 1348 (plaintiff's "attempt to limit the abstract idea ... to a particular technological environment" is "insufficient to save a claim in this context"); *see also Alice*, 134 S.Ct. at 2358; *Ultramercial*, 772 F.3d at 716; *buySAFE*, 765 F.3d at 1355.

Indeed, the Federal Circuit has held claims with implementation details and hardware components more specific than anything in the '500 Patent to be ineligible. *See Accenture*, 728 F.3d at 1343-45, 1338 (claims patent-ineligible despite reciting multiple specific computer components); *Bancorp*, 687 F.3d at 1274-78 (same). Like the patents in these cases, FairWarning's patent fails to claim anything that transforms it from anything other than an invalid abstract idea. In sum, FairWarning's patent claims, like those in *Ultramercial*, "were recited too broadly and generically to be considered sufficiently specific

and meaningful applications of their underlying abstract ideas." *DDR Holdings*, 2014 WL 6845152 at *9.

IV. CONCLUSION

For the foregoing reasons, Iatric respectfully requests that the Court dismiss FairWarning's Amended Complaint because all of the claims of the asserted patent are directed to patent-ineligible subject matter.

REQUEST FOR ORAL ARGUMENT

Pursuant to Local Rule 3.01 (j), Iatric respectfully requests the Court schedule oral argument on Iatric's motion. Iatric estimates that approximately 15-20 minutes per side is needed for argument.

CERTIFICATE OF SERVICE

I certify that on April 6, 2015, I electronically filed the foregoing with the Clerk of the Court by using the CM/ECF system which will send a notice of electronic filing to all counsel of record.

/s/ Richard E. Fee
Richard E. Fee
Florida Bar No. 813680
Kathleen M. Wade
Florida Bar No. 127965
FEE & JEFFRIES, P.A.
1227 N. Franklin Street
Tampa, Florida 33602
(813) 229-8008
(813) 229-0046 (Facsimile)
rfee@feejeffries.com
kwade@feejeffries.com

Local Counsel for Iatric Systems, Inc.

Lisa M. Tittemore
Brandon T. Scruggs
Admitted pro hac vice
SUNSTEIN KANN MURPHY & TIMBERS LLP
125 Summer Street
Boston, MA 02110-1618
(617) 443-9292
(617) 443-0004 (Facsimile)
ltittemore@sunsteinlaw.com
bscruggs@sunsteinlaw.com

Trial Counsel for Iatric Systems, Inc.

EXHIBIT A

US008578500B2

(12) United States Patent
Long

(10) Patent No.: US 8,578,500 B2
(45) Date of Patent: Nov. 5, 2013

(54) **SYSTEM AND METHOD OF FRAUD AND MISUSE DETECTION**

(76) Inventor: **Kurt James Long**, St. Petersburg, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 734 days.

(21) Appl. No.: **11/687,864**

(22) Filed: **Mar. 19, 2007**

(65) **Prior Publication Data**

US 2007/0220604 A1 Sep. 20, 2007

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/420,645, filed on May 26, 2006.

(60) Provisional application No. 60/685,655, filed on May 31, 2005.

(51) **Int. Cl.**
G06F 21/00 (2013.01)

(52) **U.S. Cl.**
USPC **726/26**; 726/22

(58) **Field of Classification Search**
USPC 726/22, 26
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,557,742	A	9/1996	Smaha et al.	
6,347,374	B1	2/2002	Drake et al.	
6,405,318	B1	6/2002	Rowland	
6,549,208	B2	4/2003	Maloney et al.	
6,789,202	B1	9/2004	Ko et al.	
2003/0229519	A1 *	12/2003	Eidex et al.	705/2
2004/0260945	A1	12/2004	Raikar et al.	
2005/0027848	A1 *	2/2005	Kamenetsky et al.	709/223
2006/0282660	A1 *	12/2006	Varghese et al.	713/155
2007/0039049	A1	2/2007	Kupferman et al.	

FOREIGN PATENT DOCUMENTS

EP	1054529	A2	11/2000

OTHER PUBLICATIONS

"Security Management" eTrust® Audit. 2006. http://www3.ca.com/solutions/Product.aspx?ID=157.

Denning D E: "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, IEEE Service Center, Los Alamitos, CA, vol. SE-13, No. 2, (Feb. 1, 1987)<pp. 222-232.

Debar et al.: "Towards a taxonomy of intrusion-detection systems," Computer Networks, Elsevier Science Publishers B.V., Amsterdam, NL, vol. 31, No. 8, (Apr. 23, 1999), pp. 805-822.

European Search Report for 08 743 964.2 dated May 9, 2011.

* cited by examiner

Primary Examiner — Lisa Lewis

(74) *Attorney, Agent, or Firm* — Lowe Hauptman Ham & Berner, LLP

(57) **ABSTRACT**

A system and method are provided for detecting fraud and/or misuse of data in a computer environment through generating a rule for monitoring at least one of transactions and activities that are associated with the data. The rule can be generated based on one or more criteria related to the at least one of the transactions and the activities that is indicative of fraud or misuse of the data. The rule can be applied to the at least one of the transactions and the activities to determine if an event has occurred, where the event occurs if the at least one criteria has been met. A hit is stored if the event has occurred and a notification can be provided if the event has occurred. A compilation of hits related to the rule can be provided.

17 Claims, 10 Drawing Sheets

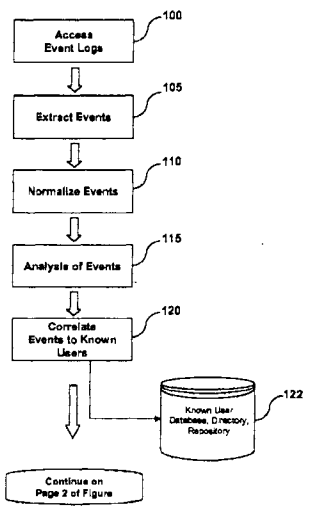


Figure 1A

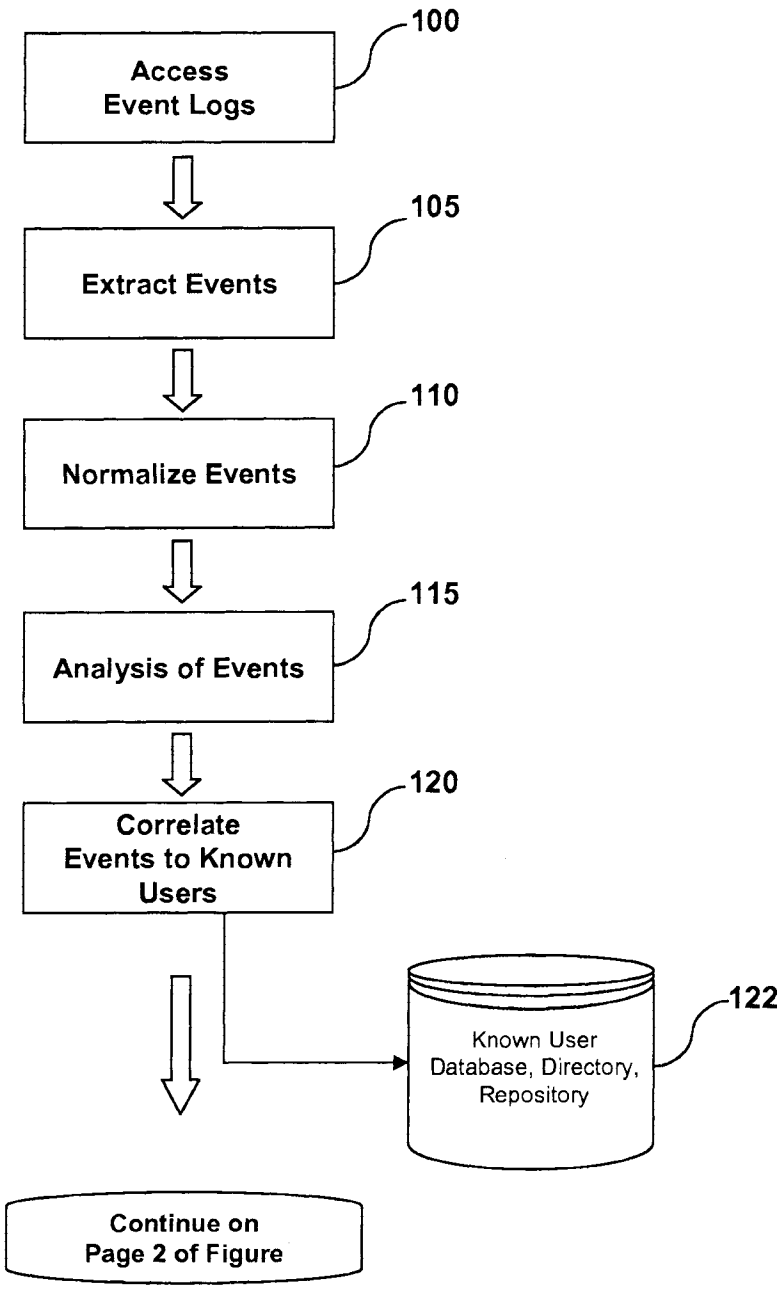

100
Access Event Logs
105
Extract Events
110
Normalize Events
115
Analysis of Events
120
Correlate Events to Known Users
122
Known User Database, Directory, Repository
Continue on Page 2 of Figure

Figure 1B

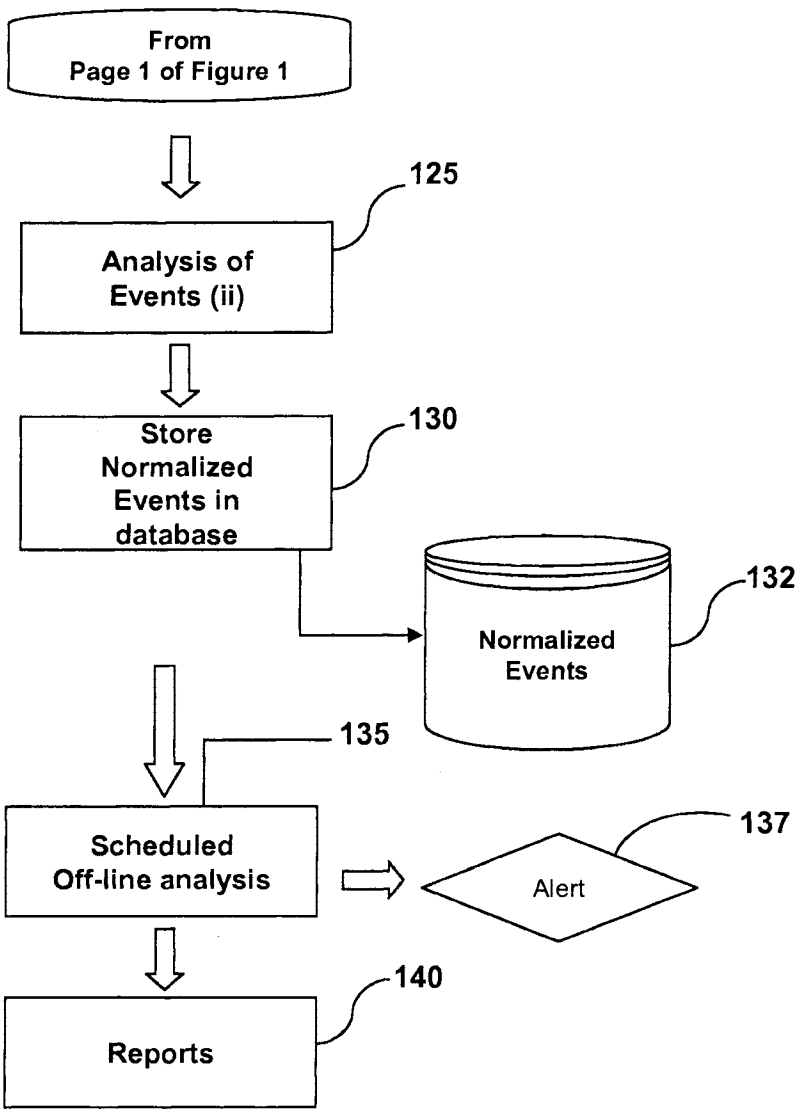

From
Page 1 of Figure 1
125
Analysis of
Events (ii)
130
Store
Normalized
Events in
database
132
Normalized
Events
135
Scheduled
Off-line analysis
137
Alert
140
Reports

Figure 2

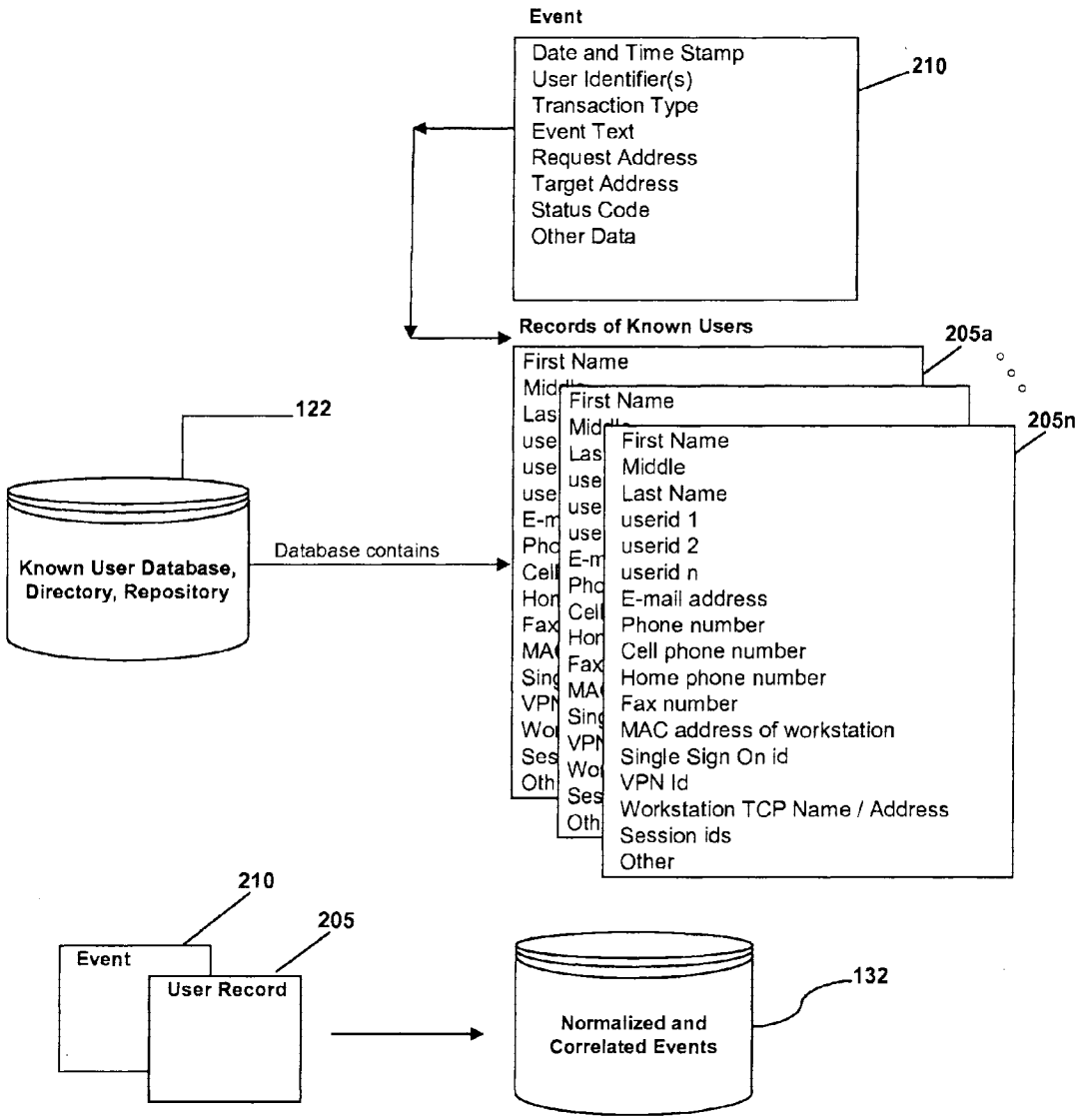
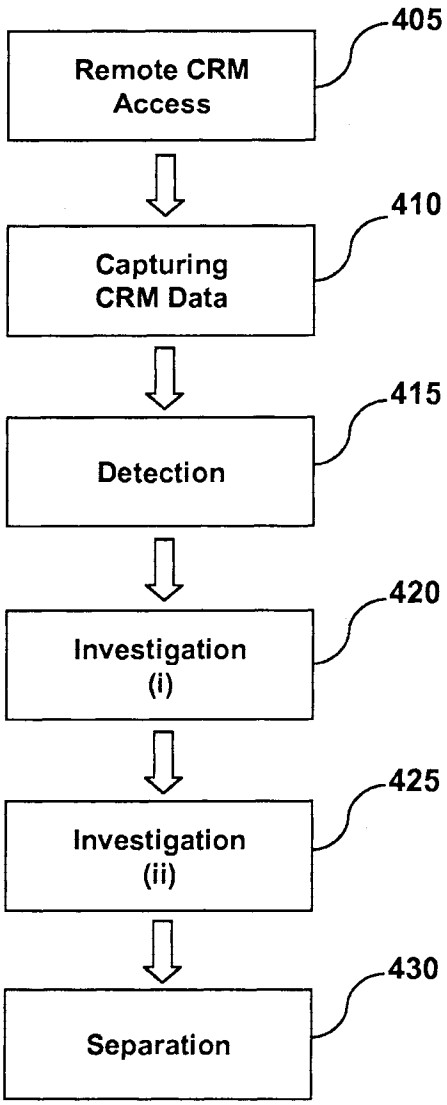

Event
Date and Time Stamp
User Identifier(s)
Transaction Type
Event Text
Request Address
Target Address
Status Code
Other Data
210
Records of Known Users
205a
205n
First Name
Middle
Last Name
userid 1
userid 2
userid n
E-mail address
Phone number
Cell phone number
Home phone number
Fax number
MAC address of workstation
Single Sign On id
VPN Id
Workstation TCP Name / Address
Session ids
Other
122
Known User Database, Directory, Repository
Database contains
210
205
Event
User Record
Normalized and Correlated Events
132

Figure 3

```
<?xml version="1.0" encoding="UTF-8" ?>
- <LogFormatDefinition definitionName="SharePoint" compatibleWith="web">
- <event numLines="1">
- <timeStampFields>
<numTSFields>2</numTSFields>
<concatWith />
<fmtString>yyyy-mm-ddHH:mm:ss</fmtString>
<tsField>date</tsField>
<tsField>time</tsField>
</timeStampFields>
- <parseRules>
- <!-- rules for parsing whole lines of data like ignore rules -->
- <rule ruleType="ignore" constraint="startsWith">
- <!-- constraint values: startsWith, endsWith, contains -->
<searchString text="comment line">#</searchString>
- <!-- ignore lines that start with # because they are comment lines -->
</rule>
</parseRules>
- <field fieldName="date" parseType="delimited">
- <!-- parseType values: delimited, bounded, indexed -->
<delimiter />
<delimitedIndex>0</delimitedIndex>
</field>
- <field fieldName="time" parseType="delimited">
<delimiter />
<delimitedIndex>1</delimitedIndex>
</field>
- <field fieldName="serverip" parseType="delimited">
<delimiter />
<delimitedIndex>2</delimitedIndex>
</field>
- <field fieldName="method" parseType="delimited">
<delimiter />
<delimitedIndex>3</delimitedIndex>
</field>
- <field fieldName="uri" parseType="delimited">
<delimiter />
<delimitedIndex>4</delimitedIndex>
</field>
- <field fieldName="query" parseType="delimited">
<delimiter />
<delimitedIndex>5</delimitedIndex>
</field>
- <field fieldName="username" parseType="delimited">
<delimiter />
<delimitedIndex>7</delimitedIndex>
</field>
- <field fieldName="clientip" parseType="delimited">
<delimiter />
<delimitedIndex>8</delimitedIndex>
</field>
- <field fieldName="httpstatus" parseType="delimited">
<delimiter />
<delimitedIndex>10</delimitedIndex>
</field>
- <field fieldName="s-port" parseType="delimited">
<delimiter />
<delimitedIndex>6</delimitedIndex>
</field>
- <field fieldName="cs(User-Agent)" parseType="delimited">
<delimiter />
<delimitedIndex>9</delimitedIndex>
</field>
</event>
</LogFormatDefinition>
```

301

Figure 4


405
Remote CRM Access
410
Capturing CRM Data
415
Detection
420
Investigation (i)
425
Investigation (ii)
430
Separation

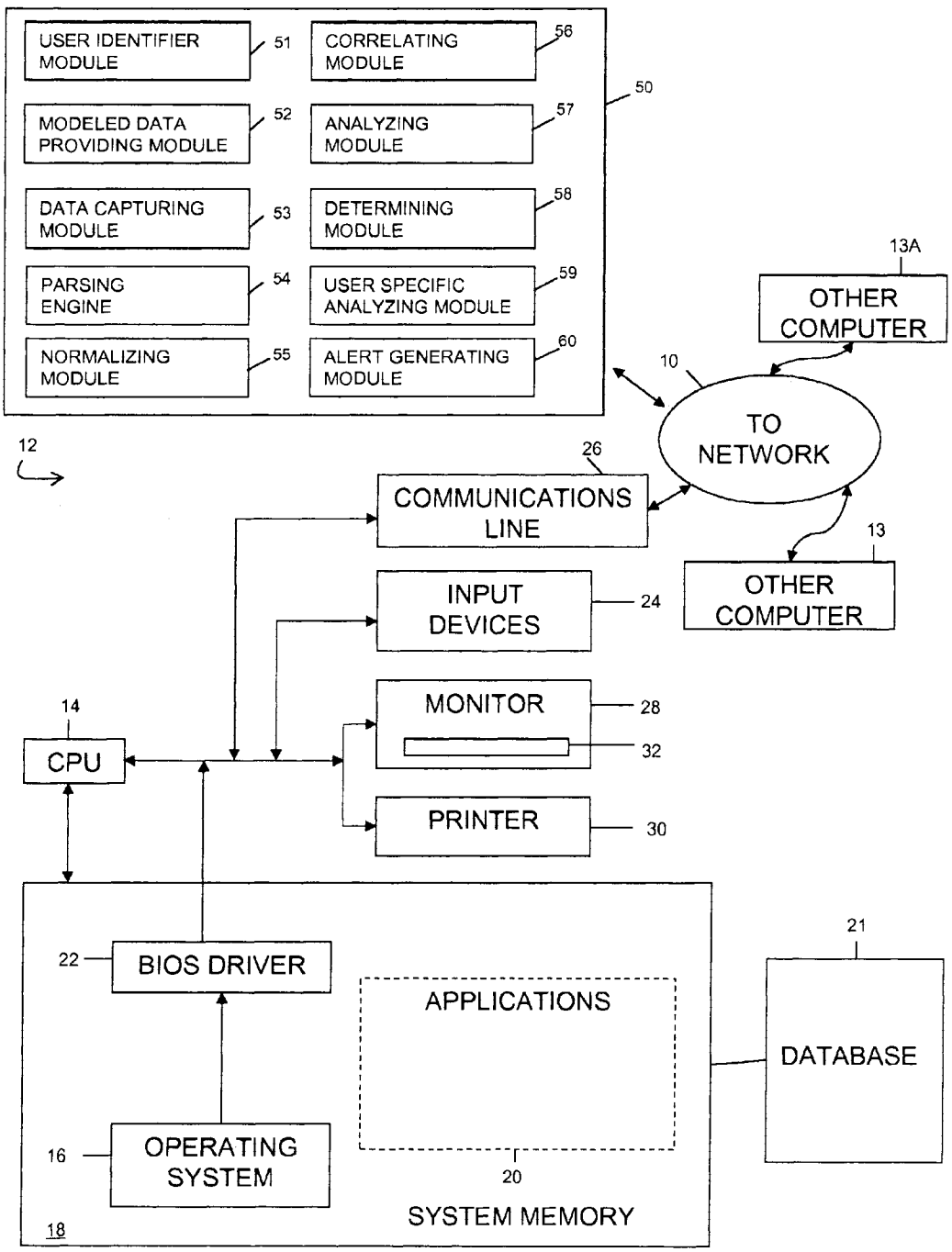

USER IDENTIFIER MODULE
51
CORRELATING MODULE
56
50
MODELED DATA PROVIDING MODULE
52
ANALYZING MODULE
57
DATA CAPTURING MODULE
53
DETERMINING MODULE
58
PARSING ENGINE
54
USER SPECIFIC ANALYZING MODULE
59
NORMALIZING MODULE
55
ALERT GENERATING MODULE
60
13A
OTHER COMPUTER
10
TO NETWORK
12
26
COMMUNICATIONS LINE
13
OTHER COMPUTER
INPUT DEVICES
24
MONITOR
28
32
14
CPU
PRINTER
30
21
22
BIOS DRIVER
APPLICATIONS
DATABASE
16
OPERATING SYSTEM
20
SYSTEM MEMORY
18

Figure 5

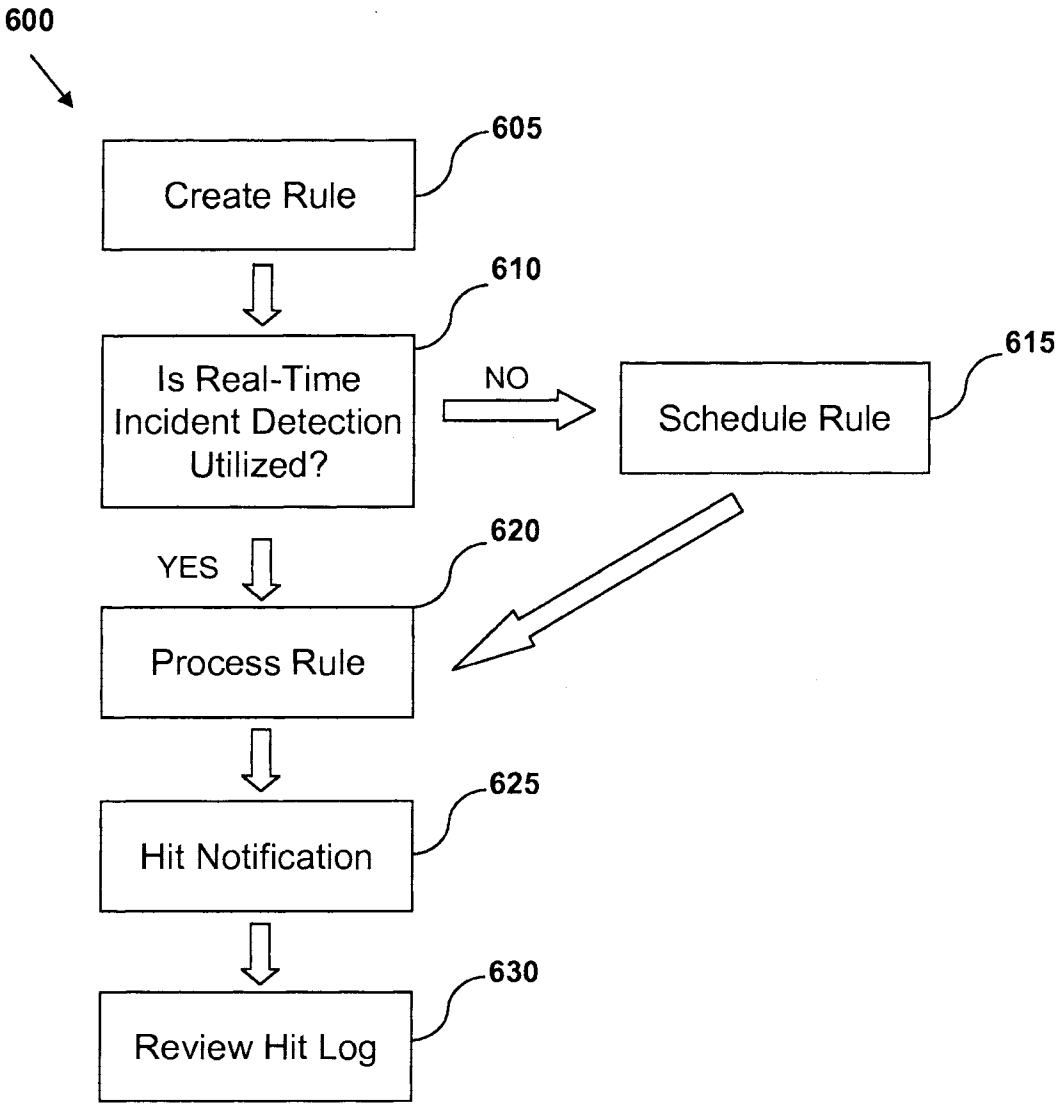

600
605
Create Rule
610
Is Real-Time Incident Detection Utilized?
NO
615
Schedule Rule
YES
620
Process Rule
625
Hit Notification
630
Review Hit Log

Figure 6

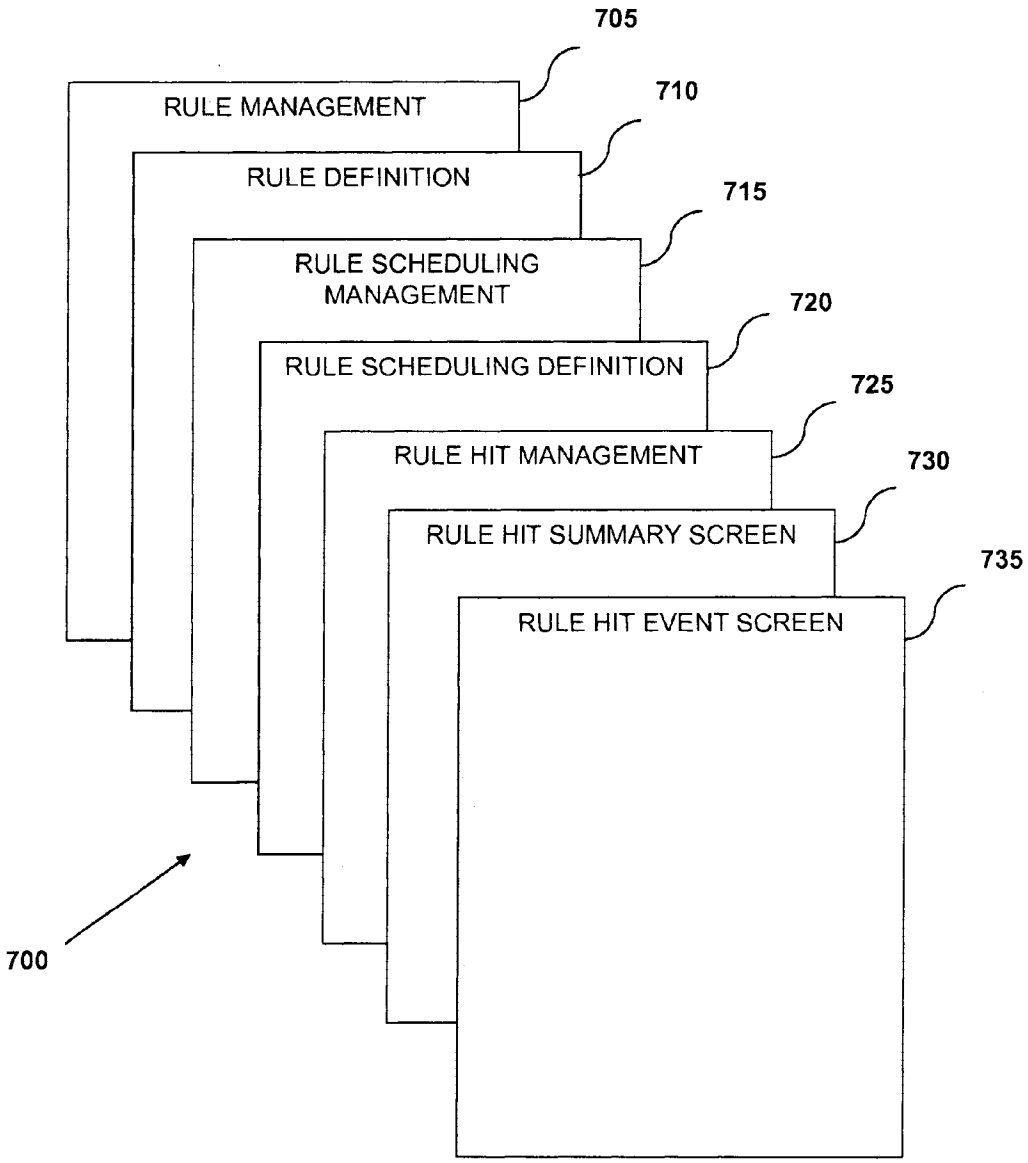

705
RULE MANAGEMENT
710
RULE DEFINITION
715
RULE SCHEDULING MANAGEMENT
720
RULE SCHEDULING DEFINITION
725
RULE HIT MANAGEMENT
730
RULE HIT SUMMARY SCREEN
735
RULE HIT EVENT SCREEN
700

Figure 7

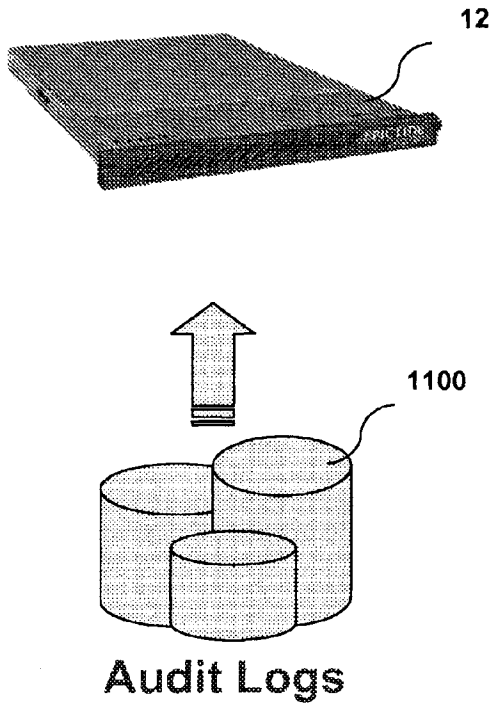


Figure 8

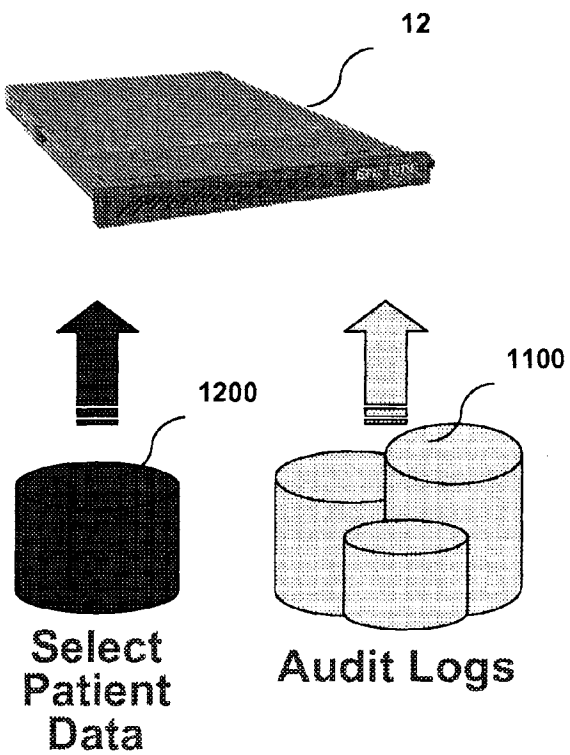


Figure 9

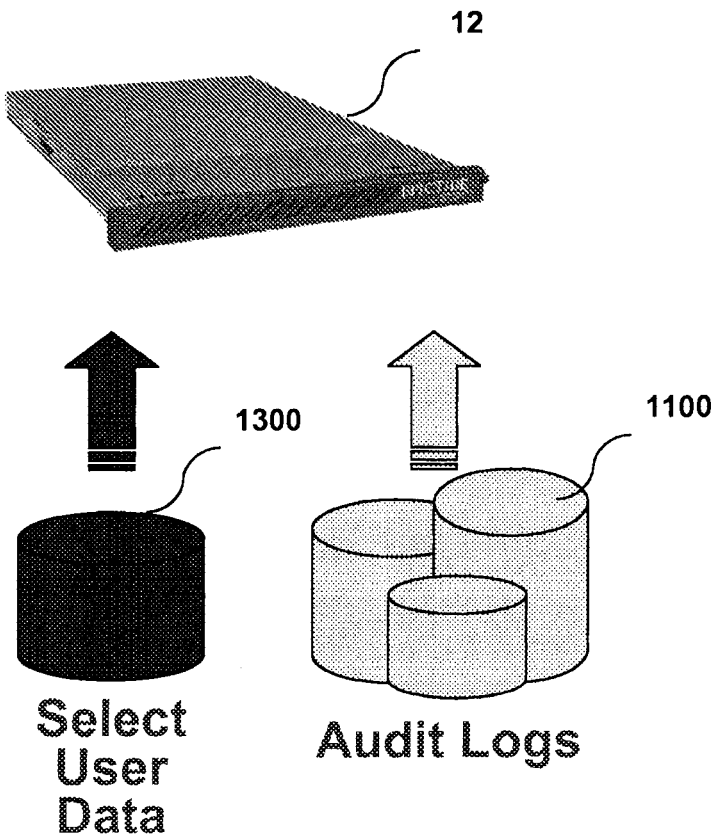


Figure 10

SYSTEM AND METHOD OF FRAUD AND MISUSE DETECTION

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of U.S. patent application Ser. No. 11/420,645, filed on May 26, 2006, which claims priority to U.S. Provisional Application Ser. No. 60/685,655, filed May 31, 2005, the entire contents of which are incorporated herein by reference.

FIELD OF THE INVENTION

The invention relates to a system and method of detecting fraud and/or misuse in a computer environment based on analyzing data such as in log files, or other similar records, including user identifier data. More particularly, the invention relates to a system and method of detecting fraud and/or misuse in a computer environment based on analyzing application layer data such as in log files, including user identifier data.

BACKGROUND OF THE INVENTION

Conventional systems for detecting fraud or misuse by users are deficient at least because conventional systems have limited abilities to recognize log file formats and access the log files. This is especially difficult when a system accesses file logs that are generated by different applications, since each application may generate a different log file format.

Other problems with conventional systems include that users may have several different ways of accessing company (or other similar organizations) systems. For example, in many instances, users may use several different user-ids and passwords to access different applications or data stores of an organization. Fraud or misuse detection systems may have no way to correlate the activity of the user across the various applications. Likewise, in some instances, evaluating the behavior of a user based on one application may not provide enough information to discern a pattern of behavior that may be indicative of fraud or misuse of a company's system or information.

Some of the prior art systems related to detecting fraud and misuse of a system are described in U.S. Pat. Nos. 5,557,742 (Method and System for Detecting Intrusion Into and Misuse of a Data Processing System), 6,347,374 (Event Detection), 6,405,318 (Intrusion Detection System), and 6,549,208 (Information Security Analysis System). Various other drawbacks exits with these systems and with other systems known in the art.

SUMMARY OF THE INVENTION

Various aspects of the invention overcome at least some of these and other drawbacks of existing systems. According to one embodiment, a system and method are provided for tracking a user across logs at an application layer of various applications that a user may access.

According to one embodiment, event log files may be accessed by a monitoring system, wherein the event log files are associated with known users of users whose identity the system can derive. The event logs may be compilations of recorded transactions and/or activities that are recorded by applications and access layer devices. According to one embodiment, the events contained in the event logs may be extracted by the monitoring system. The extracted events may

be normalized into records that are suitable for analysis, storage and/or reporting. The normalized events may be analyzed against fraud scenarios that are defined for a given environment. According to one embodiment, the events may be correlated to users of the systems and the event records may contain identifiers that correlate to known users.

According to one embodiment, the normalized and correlated events may be analyzed for user specific fraud monitoring scenarios that are modeled based on a user's specific identity or role/relationship with an organization.

According to one embodiment, a method of detecting fraud or misuse of data in a computer environment is provided. The method comprises generating a rule for monitoring at least one of transactions and activities that are associated with the data, with the rule comprising at least one criteria related to the at least one of the transactions and the activities that is indicative of fraud or misuse of the data; applying the rule to the at least one of the transactions and the activities to determine if an event has occurred, with the event occurring if the at least one criteria has been met; storing a hit if the event has occurred; providing notification if the event has occurred; and providing a compilation of hits related to the rule.

According to one embodiment, a system for detecting fraud or misuse of data in a computer environment is provided. The system comprises a user interface for selection of at least one criteria related to at least one of transactions and activities associated with the data that is indicative of fraud or misuse of the data and for selection of a schedule for application of a rule for monitoring the at least one of the transactions and the activities, and a microprocessor in communication with the user interface and having access to the transactions and the activities of the data. The microprocessor generates the rule based at least in part on the at least one criteria selected and applies the rule to the at least one of the transactions and the activities according to the schedule selected to determine if an event has occurred. The event occurs if the at least one criteria has been met. The microprocessor stores a hit if the event has occurred and provides notification if the event has occurred. The microprocessor generates a compilation of hits related to the rule.

According to one embodiment, a computer readable program embodies in an article of manufacture comprising computer readable program instructions for detecting fraud or misuse of data in a computer environment is provided. The program comprises program instructions for causing the computer to provide a selection of at least one criteria related to at least one of transactions and activities associated with the data that is indicative of fraud or misuse of the data, program instructions for causing the computer to generate a rule based at least in part on the at least one criteria for monitoring the at least one of the transactions and the activities; program instructions for causing the computer to provide a selection for a schedule for application of the rule to the at least one of the transactions and the activities; program instructions for causing the computer to apply the rule according to the schedule selected to the at least one of the transactions and the activities to determine if an event has occurred, with the event occurring if the at least one criteria has been met, program instructions for causing the computer to store a hit if the event has occurred; program instructions for causing the computer to provide notification if the event has occurred; and program instructions for causing the computer to provide a compilation of hits related to the rule.

The invention has numerous advantages over and avoids many drawbacks of prior systems. These and other objects, features and advantages of the invention will be apparent through the detailed description of the embodiments and the

drawings attached thereto. It is also to be understood that both the foregoing general description and the following detailed description are exemplary and not destructive of the scope of the invention. Numerous other objects, features and advantages of the invention should now become apparent upon a reading of the following detailed description when taken in conjunction with the accompanying drawings, a brief description of which is included below.

DESCRIPTION OF THE DRAWINGS

FIGS. **1**A and **1**B illustrate a flow chart of a process flow according to one embodiment of the invention.

FIG. **2** illustrates one process of correlating events to known users according to one embodiment of the invention.

FIG. **3** illustrates exemplary XML definitions according to one embodiment of the invention that may be used for event parsing.

FIG. **4** illustrates a flow diagram of fraud detection according to one embodiment of the invention.

FIG. **5** illustrates a general purpose computing system that is connected to a network that may be used to implement one or more aspects of the monitoring system.

FIG. **6** illustrates a flow diagram of fraud or misuse detection process according to another embodiment of the invention.

FIG. **7** illustrates a user interface for a system that utilizes the process of FIG. **6**.

FIG. **8** illustrates a flow chart for detection of various fraud or misuse scenarios based upon audit logs in another embodiment of the invention.

FIG. **9** illustrates a flow chart for detection of various fraud or misuse scenarios based upon audit logs and select patient data in another embodiment of the invention.

FIG. **10** illustrates a flow chart for detection of various fraud or misuse scenarios based upon audit logs and select user data in another embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGS. **1**A and **1**B together form a flow chart that illustrate some of the processes in one embodiment of the invention. In step **100**, event log files (hereafter event logs) are accessed by a monitoring system that is provided by the invention. According to one embodiment, event logs are data stores containing events, associated with known users, that are accessed by the system from servers and devices on a network. According to an alternative embodiment of the invention, event logs may include temporary storage devices. According to another embodiment, event logs may be sent to the monitoring system via protocols and message sets. Whether accessed on servers or received via messages, the monitoring system accesses event logs associated with known users or users whose identity the system can derive.

According to one embodiment, the event logs may be compilations of recorded transactions and/or activities that are recorded by applications and access layer devices. According to one embodiment, these may include servers and applications such as VPN devices, third party applications, in-house applications, web servers, single sign on servers, databases, e-mail servers, print servers, fax servers, phone systems and any other device or server that contains or generates event information based on a known user's use or interaction with an organization's information systems. The collection of data from the event logs is scheduled by the monitoring system to be conducted periodically or performed in real-time as the events are generated.

According to one embodiment, in operation **105**, the events that are contained in the event logs may be extracted by the monitoring system using, for example, a parsing engine. According to one embodiment, the parsing engine may be an application that is configurable, for example, by using XML templates. According to one embodiment, the parsing engine maintains XML templates (as an example of standard format for a known event) of known event logs and events. The XML templates also may contain information that identifies correlations between events and event logs and may further contain information on what is to be extracted from the event for subsequent analysis, storage and reporting. For example, the XML template may contain the format of the data contained in an event log so that the data in the event log may be easily correlated to known fields based on the XML template information. One skilled in the art would recognize that XML templates are one embodiment of such a template and other similar templates or mapping techniques could also be used as would be recognized by those skilled in the art. For never previously encountered event data formats, the parsing engine may be configured via manual definition and manipulation of a default XML template to create a suitable XML template, or configured via a tool with a graphical user interface to define the event formed as would be within the abilities of one skilled in the art.

According to one embodiment, in operation **110**, the extracted events may be normalized (using, for example, the above described templates) into records that are suitable for analysis, storage and reporting. As part of the normalization process, an event source identifier (or event log identifier), date/time, source network address, destination network address, text associated with the event, and transaction code may be placed into the record. Based on the source identifier, additional information may be stored in the record that may not be part of a standard normalized record. For example, the record may include information correlating the events to the event source identifiers. One skilled in the art would recognize that the fields listed here are exemplary only and those skilled in the art would recognize various alternatives and modifications all of which are considered as a part of the invention.

According to one embodiment, in operation **115**, the normalized events may be analyzed against fraud scenarios that are defined for a given organizational environment. Examples of such analysis include monitoring for access to a specific type of record in a healthcare financial service or mortgage environment, or monitoring for a volume of transactions over a specified time period. Alerting and off-line reports may be generated by the system. This stage of analysis is characterized by analyzing for scenarios that benefit from being detected rapidly. The analysis of fraud scenarios is discussed in greater detail further herein.

According to one embodiment, in operation **120**, events may be correlated to users of the organization's systems. According to one embodiment, the event records may contain identifier(s) that correlate to known users. The listing of identifiers that identify a user may be sorted or accessible in a data repository **122**, as will be discussed in further detail further herein. These correlation identifiers (found in the event records) may include e-mail address, userid(s), database ids, phone number, session id, TCP/IP address, MAC address, single sign on id, or any other id (identifier) that may correlate uniquely to users in a given organization's environment. According to one embodiment, these identifiers may be placed into the normalized record, such that the normalized records are associated with known users. Using the identifier, the monitoring system may correlate the normalized events

using a database, directory or general repository **122** of known users. According in one embodiment, events that can not be matched against known users (for example, users that cannot be identified based on the known users in the repository **122**) may be maintained in a separate records list. According to another embodiment, attempts to match the records to known users may be performed in an off-line process which may be performed later in time or which may be initiated in near real-time by the monitoring system sending a message to initiate the matching of the unknown record. According to one embodiment, the monitoring system is capable of maintaining its own user repository **122**. According to another embodiment, the monitoring system is capable of interfacing with an identity management repository, a single sign on repository, a human resource repository, a ERP or any other repository of known users. Alternatively, the monitoring system may use a combined approach in which it first checks it own repository **122** before interfacing the other repositories of user information in an organization.

According to one embodiment, in operation **125**, the normalized and correlated events may be analyzed using, for example, rules, algorithms, database queries, or other methods, for user specific fraud monitoring scenarios that are modeled based on a user'specific identity or role relationship with an organization. According to one embodiment, the fraud scenarios may be modeled and stored in XML templates. For example, monitoring system may include a template that is matched to determine whether a fraud or misuse scenario has arisen. Examples of fraudulent and misuse scenarios are discussed further herein.

According to one embodiment, in operation **132**, the normalized and correlated events may be stored in a database **132** for subsequent analysis and reporting. According to one embodiment, events that are non-correlated with users may be maintained in a separate records list and attempts to match the records to known users may be performed in an off-line process.

According to one embodiment, in operation **135**, the monitoring system may analyze the off-line database of normalized and correlated events **132** for fraud scenarios that can not be detected in real time due to data, time or performance limitations. The monitoring system may produce alerts **137** if its off line analysis uncovers fraudulent scenarios. These alerts may be in the form of a report or message, which alerts a responsible person to investigate the fraud or misuse scenarios. According to another embodiment, the monitoring system my initiate preventive action, for example, by suspending the access of a known user whose activities have triggered the alert. According to another embodiment, in operation **140**, the system may produce generalized security reporting based on transactions and access by authenticated users. Such reports may be used to track the security of an organization's systems or may be used for subsequent investigation, once a fraud or misuse scenario has been uncovered.

The following description provides specific embodiments for some of the operations discussed above. While specific embodiments of the invention are discussed herein and are illustrated in the drawings appended hereto, the invention encompasses a broader spectrum than the specific subject matter described and illustrated. As would be appreciated by those skilled in the art, the embodiments described herein provide but a few examples of the broad scope of the invention. Thee is no intention to limit the scope of the invention only to the embodiments described herein.

1. Accessing Events. According to one embodiment, the monitoring system is flexible in its ability to read events. According to one embodiment, an application layer protocol such as Simple Network Management Protocol (SNMP) may be used to facilitate the exchange of management information between network devices. The monitoring system simply needs programmatic input (or read) access to a given event source such as a log file. In the case of a log file, the log file may be accessible via a local hard drive, a network hard drive, and/or may be transferred locally via a file transfer protocol such as ftp. According to one embodiment, the monitoring system is also flexible enough to read from a local or remote database via protocols, such as ODBC, in order to access relevant events. Alternatively, a log file may be generated through the systematic extraction from one or more databases, and the generated log file(s) then transported via ftp to the local drive of the monitoring system. According to another embodiment, the monitoring system may provide a web service interface in order to receive events using a message protocol, such as Simple Object Access Protocol (SOAP). As previously stated, the monitoring system generally is flexible and uses programmatic (read) access to event sources.

2. Event Contents and Format. According to one embodiment, while the monitoring system is capable of processing any log event, it has the ability to process events that were directly or indirectly generated by known users (known, for example, to an organization) and then correlate those events to the known users. For user associated events, one general formed of the event data that is tracked is outlined below. Of course, it should be recognized that this format is exemplary only and those skilled in the art would recognize various modifications and alternatives all of which are considered at a part of the present invention. One general format may include [Date and Time Stamp] [User identifier] [Transaction Type] [Event Text] [Request Address] [Target Address] [Status Code] [Other Data]. Other formats are contemplated.

As should be recognized by one skilled in the art, the number of lines per event, field order, delimiters, field format, etc. may vary between applications, access servers, databases, etc. The monitoring system is sufficiently configurable to handle various events. The "User identifier" field may be a user-id, an e-main address, a phone number, a database-id, a single sign on id, a TCP/IP address, a MAC address, a session id or any other identifier that ties the event to a known user. The applicability of the identifier may be dependent on the organization's environment, including user-id policies, application environments, network layouts, etc. The monitoring system is sufficiently configurable to allow for these variables in correlating the events to known users.

3. Event Definitions. According to one embodiment, the monitoring system may be flexible in its ability to process the above described events. According to one embodiment, the system may include a XML based description language that is used to specify the variables of a given event type such as fields, field order, field delimiters number of lines per event, number of characters, field type and spoken language type. Multiple event types in a given event source (such as a log file) can also be similarly described. According to one embodiment, the definition of event types may be maintained in a directory that is known to the monitoring system so that they may be used whenever a given event type (which has a definition in the directory) is processed.

4. System Database Schemas. According to one embodiment, the monitoring system may maintain a set of schemas that correspond to the event types being processed. These schemas may be used to generate database tables. For example, "http common log format" has a pre-defined schema that the monitoring system maintains and can generally re-use whenever the events of a "http common log format" type are processed. According to another embodiment,

the monitoring system may provide the ability to use a schema that associates fields that are unique to a specific event type to the storage format of an event. In other words, the system may be sufficiently configurable to handle event fields that are not part of a standard format as described above. For example, program logic based on keywords or certain alphanumeric sequences may be used to identify fields in an event data record and may correlate them to the standardized storage format of the normalized records.

According to one embodiment, the monitoring system may normalized events by mapping as many fields available as described above to the schema and table defined herein as well as mapping the event specific fields to the table and field as described in the event type's specific schema. According to another embodiment, the monitoring system may generate a unique identifier for every event processed and stored in the system's database(s), which may be used for subsequent indexing, correlation and reporting. According to one embodiment, suitable indexed fields may be part of the schema definition that allows for increased efficiency in accessing the stored data, generating reports and in processing events. The normalized event generally may contain the same data as contained in an event record, but it may be formatted and indexed for a database.

According to one embodiment, the monitoring system may maintain tables (in a database **132**) that correspond to known users and associated identifiers for an organization. According to one embodiment, the monitoring system may be sufficiently flexible to leverage existing identity management systems for the maintenance of the users and identifiers. These systems may include directories such as Active Directory or Identity Management systems from venders such as Computer Associates, BMC, Sun, IBM, Novell. Generally, the system is flexible enough to leverage existing identity sources of all kinds or to maintain the identifies itself in a repository.

5. Known User Correlation. According to one embodiment, the monitoring system may be flexible in that, depending on the processing environment and application of the system, it may correlate events to known users in real-time as the events are processed. According to another embodiment, the system may correlate the events to known users during off-line processing. In both cases, the result is that events processed by the system are correlated to the known users of an organization and used for security reporting, fraud detection, monitoring, etc., as discussed herein.

According to one embodiment of the invention. FIG. **2** illustrates a diagram of a process for correlating events **210** to records of known users **205**. The monitoring system may produce the normalized event **210** by the general process outlined earlier herein. According to one embodiment, the normalized event **210** may contain one or more User identifier (s), examples of which include e-main address, userid(s), database ids, phone number, TCP/IP address, MAC address, single sign on id, session id or any other id that may correlate uniquely to a user given an organization's environment.

According to one embodiment, the system may access a directory, database or other repository of users **122** and associated identifiers, examples of which are shown in the records of known users **205**. Therefore, as shown in FIG. **2**, particular users may be associated with a wide variety of identifiers. Some of these identifiers may be maintained on a permanent base while other identifiers, such as session ids, may only be maintained for a short duration, while a particular session of the user is current or has been recently created. Likewise, different variants of a particular type of identifier may also be maintained, for example, if a user has multiple e-mail addresses or multiple telephone numbers, all of these may be stored in user repository **122**.

According to one embodiment of the invention, the monitoring system may correlate an event **210** to records of a known users **205** based on matching identifier(s). According to one embodiment of the invention, event **210** and user record **205** may be linked together in a repository **132** that contains normalized and correlated events. Session ids, and similar temporary identifiers may be captured from event records and maintained so that events **210** may be correlated to a record of known users **205** even though the event **210** may not have an identifier that directly links the event **210** to the record of known users **205**. Such temporary identifiers may be maintained in the user repository **122** or as a record in some other repository which may be linked back to the known user's record in the user repository **122**. At some point in this flow, the session id (as an example of a temporary id) should have been linked to the user within some log event. For example, a VPN typically generates a session id in association with a user login event, then subsequently only "logs" session id in events associated with that user. However, the monitoring system may track the session id based on the initial user login event so that activities of the user, identified only by the session id in event logs, can also be tracked back to the specific known user.

According to another embodiment of the invention, events for which there are no correlating user records may be stored in the database under special tables that allow reporting and additional processing.

According to one embodiment of the invention. FIG. **3** provides exemplary XML definitions **301** that may be used for event parsing.

According to one embodiment of the invention, fraud and/or misuse detection may be performed through analysis of uncorrelated events. Some fraud and misuse scenarios may be detected prior to the correlation of an event to a user. This enables the monitoring system to monitor resources of an organization and generally detect behaviors that are considered high risk, before a particular user has been identified as suspicious. For example, the monitoring system may generate an alert and alert record using any of the following techniques:

- When any user, or user in a particular category, performs a certain volume of transactions or activities over a specified time interval;
- When any use, or user in a particular category, performs a pre-defined sequence of transactions or activities;
- When any user, or user in a particular category, accesses resources outside of pre-defined hours of the day;
- When any user, or user in a particular category, changes or accesses a pre-identified resource such as a database field, file, application field; and/or
- When any user, or user in a particular category, changes or accesses resources associated with a pre-identified entity such as records associated with a famous person who checked into a hospital or records that correspond to particular customers or partner.

According to another embodiment of the invention, fraud and/or misuse detection may be performed through analysis correlated events. Some fraud and misuse scenarios may be detected when events have been correlated to users. For example, the monitoring system may generate an alert and generate an alert record using any of the following techniques:

- When any user carries out activities or transactions that are outside of pre-defined characteristics of that their rela-

tionship to the organization (job function, supplier relationship, customer relationship, etc.).

When a user carries out activities or transactions that are inconsistent with the historically established behavior of that user (or a category of users to which the user belongs);

When a pre-identified user performs pre-defined activities, transactions or gains access to system;

When a user accesses resources from an address (TCP/IP, MAC, domain, other) that is inconsistent with the past accesses; and/or

When a user conducts activities or transactions that link the user to previously established auspicious users.

Examples of the Fraudulent of Business Information Systems

The fraudulent use of business information systems may take many forms, may involve variously sophisticated participants and techniques. According to one embodiment, the monitoring system may be applied to specific forms of fraud or may be used as a more general platform against more sophisticated forms of fraud. According to one embodiment, the monitoring system may perform monitoring, reporting, and/or incident research relating to fraud conducted in conjunction with known users (or user identifiers) of an organization. These fraudulent scenarios may go undetected by using the current art of firewall, intrusion detection and prevention, authentication/authorization techniques. It should be noted that these scenarios are exemplary only and one skilled in the art would recognize various alternatives and modifications all of which are considered as a part of the invention.

1. Sale of Customer Records. For many industries, knowledge of customers represents lucrative information. Long-term healthcare, mortgage, high value financial services are all example industries in which employees, partners, suppliers and other known entities may gain access to applications, databases, etc. via known user ids. Unscrupulous users may sell this information to competitors or other parties. According to one embodiment of the invention, the monitoring system may track which users are accessing which customer data to determine in advances if any misuse situation arises, for example, if a sales person is accessing information unrelated to any of his sales clients.

2. Unauthorized Disclosure to Protected Health Information. Within the healthcare field, access to Protected Health Information (PHI) is protected by law. Persons with general access to systems which have access to PHI, may act in collaboration with a third party to obtain PHI about a neighbor, a relative, a coworker, a famous person or a person of power in order to blackmail the victim or to view confidential information that is protected by law. Medicare fraud is also common practice and may include a ring of conspirators that act together to submit false or inflated claims. This scheme may require known/trusted users to falsify the systems within a care provider. According to one embodiment of the invention, the monitoring system may closely track which user is accessing data about a famous patient or track whether a group of users are accessing relevant data about one or more patients in such a manner that the combined data accessed may be misused.

3. Changing the Ship-to Address on an Order. Organizations that process orders electronically may have the "ship-to" address changed by an existing user, such as an employee. In this case, the employee may change the address to a destination where the employee may capture the order and sells the order on the open market. Typically, this act of fraud goes undetected until the original purchaser refuses to pay an invoice or complains that the order never arrived. According to one embodiment, the monitoring system may track which user's are changing the ship-to address or if a user is changing ship-to addresses on a regular basis. Correlating the events around the transaction takes many man hours using the current state of the art.

4. Departing Employee Capturing the Customer Database. Departing sales persons are well-known for obtaining an electronic or printed copy of the customer database and prospect pipeline. They may use this data in a new position which may be with a competitive firm. According to one embodiment of the invention, the system may provide reporting and general detection capabilities and may correlate application and database activity to the user in question for review. According to one embodiment of the invention, the monitoring system may track to are if a sales person is accessing a relatively large number of sales records or if a sales person is accessing the records of customers with whom the sales person has no relationship.

5. Exploiting Weak Authentication via the Corporate Extranet or VPN. Corporate Extranets and VPN's are most typically authenticated via userid and password. As a partner to the company, a known user may have access to sensitive information such as pricing, inventory levels, inventory warehouse locations, promotions, etc. If the user leaves the "partner" firm and moves to a competitive firm, the user may still use the same userid and password to gain competitive access to the sensitive information. According to one embodiment of the invention, the monitoring system may associate the userid with a particular IP address (or domain) and raise an alert if the IP address or domain is that of competitor or an entity that is not a partner firm.

6. Non-repudiation for Bond Traders. Bond traders often speculatively purchase these securities in anticipation of market movements. In the event the markets take unexpected moves, the bond traders may deny the characteristics of their electronic order. According to one embodiment of the invention, characteristics and stages of an electronic transaction may be correlated to the known user (the trader) in order to negate any such fraudulent claim by the trader.

7. Financial Insider Trading Rings. Insider trading rings may comprise many collaborators using various electronic systems including applications, e-main, phone, and/or fax. According to one embodiment of the invention, the monitoring system may be used to detect suspicious behaviors or may be used to incident investigations to identify all conspirators. A typical scenarios is for one party to receive "inside information" from an outside source via some electronic means. The first source then collaborates with others to conduct trades that generate fraudulent profits based on the ill-gotten information. According to one embodiment of the invention, the monitoring system may detect such activities at a much earlier stage than might be possible using conventional insider trading detection methods.

8. Web Services. Business information systems are often published as web services. While authentication and authorization standards are established, the same rogue users that plague traditional systems often take advantage of a published web service. According to one embodiment of the invention, the system may provide reporting and general detection capabilities and may correlate application and database activity to the user in question for review.

According to one embodiment of the invention, FIG. **4** illustrates operations in the use of the monitoring system to detect misuse based on the actions of a departing employee. According to one exemplary scenario, a sales person who is an employee of the Organization has accepted a comparable position with a competitive firm. The employee has not notified the Organization of their intent to leave and is continuing

to work in a business as usual appearance. The employee has decided to accumulate as many information resources as possible that may help with new business at their next position.

1. Customer and Prospect Record Access. As part of their job, the Employee has access to detailed information on the Organization's customer and prospects. Customer and prospect records are maintained in a CRM (Customer Relationship Management) application, which is available through the Organization's VPN and Extranet. The CRM application has a privilege management system for limiting access to records to the "owner of the record" only. However, due to the collaborative nature of the sales and support process, this feature is rarely used, so that all employees have access to all records.

2. Remote Data Capture. Knowing specifics on customers and prospects who are actively engaged with the Organization would be valuable in saving time and generating new business at their next position. In operation **405**, the Employee decides to access the CRM application through the corporate VPN and to capture prospects and customers of the Organization in operation **410**. The Employee's work location is in a remote office, away from the Organization's headquarters, to the Employee is comfortably able to take an entire morning accessing the CRM system to electronically capture over 125 customer and prospect records. The electronically captured customer and prospect records are then forwarded to a personal "hotmail" e-main account. The Employee intended to access another 200 records at later times.

3. Detection. According to one embodiment of the invention, the monitoring system may be configured to monitor access to CRM, VPN and Internet proxy logs. The monitoring system may be configured to alert the security team in the event that more than 50 customer or prospect records are accessed in a specific (for example, four hour) time period. Thus, actions of the departing Employee may trigger a security alert in operation **415**.

4. Investigation. According to one embodiment of the invention, in operations **420** and **425**, the monitoring system may facilitate a forensic investigation once an alert has been generated. Once the security team had been alerted of a potential incident, they can run a report from the monitoring system which has captured events from the VPN, CRM and Internet proxy from the last 30 days. According to one embodiment, from this report, the security team may be able to determine that the employee had remotely accessed 125 customer and prospect records through the corporate VPN and that the employee had also sent a series of e-mails to a hotmail account during the same time period. According to one embodiment, this analysis may be performed using automated roles to determine that a fraud/misuse situation has been detected.

According to one embodiment of the invention, the security team can then forward this information or an automated alert can be forwarded to the Human Resources department of the Organization. In operation **430**, the Organization may then be able to confront the Employee with the facts, limiting future damages and enable the Organization to work through the Employee Separation in an informed manner. Alternatively, the monitoring system may automatically disable or suspend the access of the Employee to the Organization's system, so that further damage can be prevented before the situation with the Employee can be further evaluated.

According to one embodiment of the invention, FIG. **5** illustrates the components of a computing system connected through a general purpose electronic network **10**, such as a computer network. The computer network **10** may be a virtual private network or a public network, such as the Internet. As illustrated in FIG. **5**, the computer system **12** may include a central processing unit (CPU) **14** that is connected to a system memory **18**. System memory **18** may include an operating system **16**, a BIOS driver **22**, and application programs **20**. In addition, computer system **12** may include input devices **24**, such as a mouse or a keyboard **32**, and output devices such as a printer **30** and a display monitor **28**, and a permanent data store, such as a database **21**. Computer system **12** may include a communications interface **26**, such as an Ethernet card, to communicate to the electronic network **10**. Other computer systems **13** and **13A** may also be connected to the electronic network **10**, which can be implemented as a Wide Area Network (WAN) or as an inter-network, such as the Internet.

According to one embodiment, computer system **12** may include a monitoring server **50** that implements the monitoring system or its parts discussed herein, including programmed code that implements the logic and modules discussed herein with respect to FIGS. **1-4**. One skilled in the art would recognize that such a computing system may be logically configured and programmed to perform the processes discussed herein with respect to FIGS. **1-4**. It should be appreciated that many other similar configurations are within the abilities of one skilled in the art and it is contemplated that all of these configurations could be used with the methods and systems of the invention. Furthermore, it should be appreciated that it is within the abilities of one skilled in the art to program and configure a networked computer system to implement the method steps of certain embodiments of the invention, discussed herein.

According to one embodiment, monitoring server **50** may include a user identifier module **51** that provides data corresponding to computer users, a modeled data providing module **52** that provides fraud detection information and misuse detection information, a data capturing module **53** that provides application layer data and data corresponding to transactions and activities that are associated with computer users, a parsing engine **54** that extracts application layer data and data corresponding to transactions and activities that are associated with the computer users, a normalizing engine **55** that normalizes the data extracted by the parsing engine, a correlating module **56** that correlates the normalized at, an analyzing module **57** that analyzes the correlated information and the modeled data, a determining module **58** that determines whether the correlated information corresponds to at least one of the fraud detection information and misuse detection information, a user specific analyzing module **59** that analyzes the correlated information for user specific fraud detection information based on the computer users identity, a pre-defined role associated with each computer user, and/or a pre-defined relationship that is defined for the computer users, and an alert generating module **60** that generates alerts. It should be readily appreciated that a greater number of lesser number of modules may be used. One skilled in the art will readily appreciate that the invention may be implemented using individual modules, a single module that incorporates the features of two or more separately described modules, individual software programs, and/or a single software program.

According to one embodiment of the invention, FIG. **6** illustrates a rule engine or process **600** that enables automatic detection of incidents which may be related to fraud or misuse of data, such as violations of the Health Insurance Portability and Accountability Act (HIPAA), identity theft and medical identity theft. The rule can monitor transactions and/or activities that are associated with the data, for example, accessing of the data by a user or non-user of the system storing the data. Process **600** can utilize one or more of the components described above with respect to system **12**, including the various modules for capturing, parsing, correlating, normal-

izing, analyzing and determining incidents that arise from the transactions and/or activities associated with the data of the computer environment, including the one or more databases having the data. The rule engine **600** is not intended to be limited to any particular type of computer environment or data or any particular type of fraud or misuse of the data. However, the type of data and type of fraud or misuse of the data can be a basis, at least in part, for one or more criteria of a rule for monitoring the transactions and/or activities associated with the data or computer environment.

In step **605**, a rule is created by the user and/or a third party, such as a consultant with particular knowledge as to fraud or misuse of the particular type of data. The rule can include algorithms, database queries and/or data analysis methods to define and/or detect fraud incidents and misuse incidents. Various criteria can be used for generating or creating the rule. The criteria can be related to the transactions and/or activities that are indicative of fraud or misuse of the data. For example, process **600** can create or generate a rule based on one or more of the following parameters:

- Timeframe criteria can be utilized, such as a date range or a user-friendly time concept, e.g., yesterday, last month, last quarter.
- Volume threshold criteria can be utilized based on the number of events found. The volume threshold criteria could be used in conjunction with the timeframe criteria.
- Field value matching criteria can be utilized which allows a user to select an event source, and then allows a user to select a field and a value for that field.
- Categorized field value matching criteria can be utilized which allows a user to choose a category and a pattern to match.
- Common user name matching criteria can be utilized which allows a user to select a common user name to be searched across all supported applications. The common user name matching criteria can be implemented where the user data for each application is imported.

Step **605** also allows a user to designate the criteria related to the notice or alert that can be used when a rule is triggered. In one embodiment, an email address of the entity to be notified of the triggering of the rule can be designated. Process **600** can use the email address of rule creator as a default for the alert. In one embodiment, the type of notice can be designated such as text to be sent in an email so the user will know which rule was tripped and any specific information that can be provided.

The scope of the rule can include a single event source, such as finding matches in a single system. For example, single event source rules can accept pattern matches with timeframe and/or volume threshold criteria. As another example, a rule could determine when access has been gained to a pre-determined number of medical records ever a pre-determine time interval. Such behavior can be indicative of medical identity theft. The scope of the rule can include multiple event source rules, such as finding matches across multiple systems. For example, multiple event source rules could monitor for common user names or access to particular data categories.

In step **610**, it can be determined whether real-time incident detection is being implemented by process **600**. Real-time incident detection processes the rule as each event is read and before insertion into a database. Process **600** can apply real-time incident detection to some, most or all of the rules that have been created in step **600**.

In step **615**, any rule that is not subject to real-time incident detection can be scheduled for processing. The schedule can be time-based and/or can utilize other factors for determining the schedule, such as system activity. The particular schedule can be related to the criteria of the rule. For example, a rule that monitors access to a pre-determined volume of medical records over a pre-determined time period may be scheduled to be processed at intervals of the pre-determined time period. An example of an application that can be used to schedule the rule is Quartz.

The present disclosure also contemplates adjustable or dynamic scheduling of the rule. A user can designate one or more criteria for scheduling the rule and the schedule can be built and thereafter automatically adjusted based upon the one or more criteria. For example, a time interval between processing of the same rule can be adjusted based upon such factors as system activity or the amount of accessible data.

In step **620**, the rule can be implemented or processed. Any rule that finds one or more matches can create a database entry, such as in a database of system **12** described above with respect to FIG. **5**, indicating a hit or triggering of the rule. The hit also can cause the notice or alert to be generated and sent to the designated recipient as in step **625**.

Based upon the receipt of the alert or notice, a user can access system **12** for additional information pertaining to the rule or plurality of rules that has been triggered as in step **630**. The additional information can provide the specific time of triggering the rule, as well as all other times the rule was triggered. A specific link can be provided in the notice or alert so that the user is brought directly to the relevant information pertaining to the hit when accessing system **12**.

In one embodiment of the invention, FIG. **7** illustrates a user interface **700** for the rules process **600**. A rule management page or window **705** can indicate to a user all of the defined rules. The rule management page **705** can also be used by the user for creating, modifying or deleting rules. A rule definition page or a window **710** can also be used for inputting information to define a new rule. A rule scheduling management page or window can indicate to a user all of the schedules of the rules. The rule scheduling management page **715** can also be used to create new schedules, modify existing schedules, and/or delete schedules. A rule schedule definition page or window **720** can be used to define the schedule for the rule to run.

A rule hit management page or window **725** can indicate to a user all rules that have had matches and the number of matches per rule. A rule hit summary screen or window **730** can indicate to a user all the entries in the database for hits for a particular rule. The rule hit summary screen **730** can show the date that the rule was triggered and the actual events that caused the rule to trigger. A rule hit event screen or window **735** can indicate to a user the one or more events that caused the rule to trigger. Manipulation between the pages or windows and between information on those pages or windows can occur by various techniques including drill-down menus and new windows. The present disclosure contemplates use of the same window for each of the functions described above.

Referring to FIG. **8**, the system **12** or a module thereof can be used in combination with audit logs **1100** for detection of various fraud or misuse scenarios. For example, the audit logs **1100** can be analyzed based upon various criteria is described above to detect employee self-examination, family member snooping, VIP snooping, snooping on co-workers who are patients, snooping whole other family (neighbors, etc). The criteria can include a high volume of billing/contact modifications, a high volume of downloading/printing functions, "Break-the-glass" functions, high activity levels for patients or users in a timeframe and/or unusual login activity. One of

ordinary skill in the art can use other criteria and other combinations of criteria for detecting fraud and misuse based upon the audit logs.

Referring to FIG. **9**, the system **12** or a module thereof can be used in combination with audit logs **1100** and select patient data **1200** for detection of various fraud or misuse scenarios. For example, the audit logs **1100** and select patient data **1200** can be analyzed based upon criteria including accessing patients who were discharged over a year ago or other specified time period or a patient who normally goes to the doctor once a year and suddenly goes 25 times in a year or some other unusual number of times.

Referring to FIG. **10**, the system **12** or a module thereof can be used in combination with audit logs **1100** and select user data **1300** for detection of various fraud or misuse scenarios. For example, the audit logs **1100** and select user data **1300** can be analyzed based upon criteria including remote physician staff accessing patients that aren't under their physician's care, accessing patients outside of their normal work area, accessing patients outside of their normal work shift or non-payroll user accessing payroll functions. Other criteria can also be used including patients with highest activity levels in a timeframe, users with highest activity levels in a timeframe, users with unusually long login sessions, users with high numbers of login failures and specific functions like blood type modifications.

As noted above, embodiments within the scope of the invention include program products comprising computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media which can be accessed by a general purpose or special purpose computer. By way of example, such computer-readable media can comprise RAM, ROM, EPROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a computer-readable medium. Thus, any such connection may be properly termed a computer-readable medium. Combinations of the above are also be included within the scope of computer-readable media. Computer-executable instructions may include, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions.

The invention is described in the general context of operational steps which may be implemented in one embodiment by a program product including computer-executable instructions, such as program code, executed by computers in networked environments. Generally, program code may include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of program code for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represent examples of corresponding acts for implementing the functions described in such steps.

The present invention in some embodiments, may be operated in a networked environment using logical connections to one or more remote computers having processors. Logical connections may include a local area network (LAN) and a wide area network (WAN) that are presented here by way of example and not limitation. Such networking environments are commonplace in office-wide or enterprise-wide computer networks, intranets and the Internet. Those skilled in the art will appreciate that such network computing environments will typically encompass many types of computer system configurations, including personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by local and remote processing devices that are linked (either by hardwired links, wireless links, or by a communication of hardwired or wireless links) through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Other embodiments of the invention will be apparent to those skilled in the art from a consideration of the specification and the practice of the invention disclosed herein. It is intended that the specification be considered as exemplary only, with the true scope and spirit of the invention also being indicated by the disclosure herein and equivalents thereof.

What is claimed is:

1. A method of detecting improper access of a patient's protected health information (PHI) in a computer environment, the method comprising:

generating a rule for monitoring audit log data representing at least one of transactions or activities that are executed in the computer environment, which are associated with the patient's PHI, the rule comprising at least one criterion related to accesses in excess of a specific volume, accesses during a pre-determined time interval, accesses by a specific user, that is indicative of improper access of the patient's PHI by an authorized user wherein the improper access is an indication of potential snooping or identity theft of the patient's PHI, the authorized user having a pre-defined role comprising authorized computer access to the patient's PHI;

applying the rule to the audit log data to determine if an event has occurred, the event occurring if the at least one criterion has been met;

storing, in a memory, a hit if the event has occurred; and

providing notification if the event has occurred.

2. The method of claim **1**, further comprising:

normalizing the audit log data to be correlated with known fields based on template information.

3. The method of claim **1**, further comprising:

obtaining role information of the authorized user; and

wherein the generated rule is based on a user's specific role.

4. The method of claim **1**, wherein application of the rule to the audit log data comprises determining a misuse of the patient's PHI by tracking access by the authorized user of patient's PHI of another person.

5. The method of claim **4**, wherein the access tracked comprises access by the authorized user in excess of a specific volume of the patient's PHI of the another person.

6. The method of claim **5**, wherein the access tracked further comprises access over a predetermined time interval.

7. The method of claim **4**, wherein the criterion related to the audit log data comprises a relation between the authorized user and the another person sufficient to detect at least one of family member snooping, VIP snooping, co-worker snooping, whole other family snooping.

8. The method of claim 1, further comprising:

accessing at least one of select patient's PHI or select user data; and

applying the rule further comprises applying the rule to at least one of the select patient's PHI or the select user data.

9. The method of claim 8, wherein applying the rule to the audit log data and the select patient's PHI comprises analysis of criteria including access by the authorized user of patient's PHI of the another person who was discharged from a medical facility more than a specified time period in the past.

10. The method of claim 8, wherein applying the rule to the audit log data and the select user data comprises tracking access by the authorized user of patient's PHI of the another person, where the authorized user is a remote physician staff member and the another person is a patient not under the care of the physician for whom the authorized user is a staff member.

11. The method of claim 8, wherein applying the rule to the audit log data and the select user data comprises tracking access by the authorized user of patient's PHI of the another person during a timeframe outside the normal work shift of the authorized user.

12. A system for detecting improper access of a patient's protected health information (PHI) in a health-care system computer environment, the system comprising:

a user interface for selection of at least one criterion related to accesses in excess of a specific volume, accesses during a pre-determined time interval, accesses by a specific user, representing at least one of transactions or activities associated with the patient's PHI that is indicative of improper access of the patient's PHI within the health-care system computer environment by an authorized user wherein the improper access is an indication of potential snooping or identity theft of the patient's PHI, the authorized user having a pre-defined role comprising authorized computer access to the patient's PHI, and for selection of a schedule for application of a rule for monitoring audit log data representing at least one of the transactions or the activities;

a microprocessor in communication with the user interface and having access to the audit log data representing the transactions or the activities of the patient's PHI, the microprocessor generating the rule based at least in part on the at least one criterion selected and applying the rule to the audit log data according to the schedule selected in order to determine if an event has occurred,

wherein the event occurs if the at least one criterion has been met,

wherein the microprocessor stores a hit if the event has occurred, and

wherein the microprocessor provides notification if the event has occurred.

13. The system of claim 12, wherein application of the rule to the audit log data comprises determining a misuse of the patient's PHI by tracking access by the authorized user of patient's PHI of another person.

14. A non-transitory computer-readable medium with computer-executable instructions embodied thereon for performing a method of detecting improper access of a patient's protected health information (PHI) in a health-care system computing environment, the method comprising:

providing a selection of a criterion related to accesses in excess of a specific volume, accesses during a pre-determined time interval, accesses by a specific user, representing at least one of transactions or activities associated with the patient's PHI within the health-care system computing environment, wherein the criterion is indicative of improper access of the patient's PHI by an authorized user wherein the improper access is an indication of potential snooping or identity theft of the patient's PHI, the authorized user having a pre-defined role comprising authorized computer access to the patient's PHI;

generating a rule based at least in part on the criterion for monitoring the at least one of the transactions or the activities;

providing a selection for a schedule for application of the rule to the at least one of the transactions or the activities;

applying the rule according to the schedule selected to the at least one of the transactions or the activities to determine if an event has occurred, the event occurring if the criterion has been met;

storing a hit if the event has occurred; and

providing notification if the event has occurred.

15. The non-transitory computer-readable medium of claim 14, wherein the providing the criterion is indicative of fraudulent claims filed by the authorized user of the health-care system computing environment with authorized access to the patient's PHI.

16. The non-transitory computer-readable medium of claim 14, wherein application of the rule to the audit log data comprises determining a misuse of the patient's PHI by tracking access by the authorized user of patient's PHI of another person.

17. The non-transitory computer-readable medium of claim 16, wherein the access tracked comprises access by the authorized user to a predetermined number of patient's PHI of the another person.

* * * * *

EXHIBIT B

Exhibit B:
Comparison of '500 Patent Claim 1 with Patent-Ineligible Claims from *Alice, Ultramercial*

FairWarning '500 Patent	**Ineligible '479 Patent in *Alice***	**Ineligbile '545 Patent in *Ultramercial***
Claim 1: A method of detecting improper access of a patient's protected health information (PHI) in a computer environment, the method comprising: generating a rule for monitoring audit log data representing at least one of transactions or activities that are executed in the computer environment, which are associated with the patient's PHI, the rule comprising at least one criterion related to accesses in excess of a specific volume, accesses during a pre-determined time interval, accesses by a specific user, that is indicative of improper access of the patient's PHI by an authorized user wherein the improper access is an indication of potential snooping or identity theft of the patient's PHI, the authorized user having a pre-defined role comprising authorized computer access to the patient's PHI; applying the rule to the audit log data to determine if an event has occurred, the event occurring if the at least one criterion has been met; storing, in a memory, a hit if the event has occurred; and providing notification if the event has occurred.	Claim 33: A method of exchanging obligations as between parties, each party holding a credit record and a debit record with an exchange institution, the credit records and debit records for exchange of predetermined obligations, the method comprising the steps of: (a) creating a shadow credit record and shadow debit record for each stakeholder party to be held independently by a supervisory institution from the exchange institutions; (b) obtaining from each exchange institution a start-of-day balance for each shadow credit record and shadow debit record; (c) for every transaction resulting in an exchange obligation, the supervisory institution adjusting each respective party's shadow credit record or shadow debit record, allowing only these transactions that do not result in the value of the shadow debit record being less than the value of the shadow credit record at any time, each said adjustment taking place in chronological order; and (d) at the end-of-day, the supervisory institution instructing on[e] of the exchange institutions to exchange credits or debits to the credit record and debit record of the respective parties in accordance with the	Claim 1: A method for distribution of products over the Internet via a facilitator, said method comprising the steps of: a first step of receiving, from a content provider, media products that are covered by intellectual property rights protection and are available for purchase, wherein each said media product being comprised of at least one of text data, music data, and video data; a second step of selecting a sponsor message to be associated with the media product, said sponsor message being selected from a plurality of sponsor messages, said second step including accessing an activity log to verify that the total number of times which the sponsor message has been previously presented is less than the number of transaction cycles contracted by the sponsor of the sponsor message; a third step of providing the media product for sale at an Internet website; a fourth step of restricting general public access to said media product; a fifth step of offering to a consumer access to the media product without charge to the consumer on the precondition that the consumer views the sponsor message;

Exhibit B:
Comparison of '500 Patent Claim 1 with Patent-Ineligible Claims from *Alice, Ultramercial*

FairWarning '500 Patent	**Ineligible '479 Patent in *Alice***	**Ineligbile '545 Patent in *Ultramercial***
	adjustments of the said permitted transactions, the credits and debits being irrevocable, time invariant obligations placed on the exchange institutions.	a sixth step of receiving from the consumer a request to view the sponsor message, wherein the consumer submits said request in response to being offered access to the media product; a seventh step of, in response to receiving the request from the consumer, facilitating the display of a sponsor message to the consumer; an eight step of, if the sponsor message is not an interactive message, allowing said consumer access to said media product after said step of facilitating the display of said sponsor message; a ninth step of, if the sponsor message is an interactive message, presenting at least one query to the consumer and allowing said access to said media product after receiving a response to said at least one query; a tenth step of recording the transaction event to the activity log, said tenth step including updating the total number of times the sponsor message has been presented; and an eleventh step of receiving payment from the sponsor of the sponsor message displayed.

EXHIBIT C

US006134664A

United States Patent [19]

Walker

[11] Patent Number: 6,134,664

[45] Date of Patent: Oct. 17, 2000

[54] **METHOD AND SYSTEM FOR REDUCING THE VOLUME OF AUDIT DATA AND NORMALIZING THE AUDIT DATA RECEIVED FROM HETEROGENEOUS SOURCES**

[75] Inventor: **Jeffrey H. Walker**, Papillon, Nebr.

[73] Assignee: **PRC Inc.**, McLean, Va.

[21] Appl. No.: **09/109,866**

[22] Filed: **Jul. 6, 1998**

[51] **Int. Cl.**[7] **G06F 13/00**

[52] **U.S. Cl.** **713/201**

[58] **Field of Search** 713/200, 201, 713/202; 709/229; 380/3, 200, 4, 201, 35, 202, 30, 203, 51

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,557,742	9/1996	Smaha et al.	713/200
5,561,795	10/1996	Sarkar .	
5,745,753	4/1998	Mosher, Jr. .	
5,778,076	7/1998	Kara et al.	380/51
5,987,611	11/1999	Freund	713/201

OTHER PUBLICATIONS

"DID (Distribution Intrusion Detection System)—Motivation, Architecture, and An Early Prototype", S.R. Snapp, et al., Proc. 14th Nat'l Computer Security Conf., Washington, D.C. (Oct. 1991), pp. 167–176.

"An Intrusion–Detection Model", D. Denning, IEEE Transactions on Software Engineering, vol. SE–13, No. 2, Feb. 1987.

Primary Examiner—Robert W. Beausoliel, Jr.
Assistant Examiner—Pierre Eddy Elisca
Attorney, Agent, or Firm—Lowe Hauptman Gopstein Gilman & Berner

[57] **ABSTRACT**

A method of reducing the volume of native audit data from further analysis by a misuse and intrusion detection engine is disclosed. Typically, more than ninety percent of the volume of audit information received from heterogeneous operating systems does not need to be analyzed by a misuse and intrusion detection engine because this audit information can be filtered out as not posing a security threat. Advantageously, by reducing (eliminating) the volume of audit information, a misuse and intrusion engine can more quickly determine whether a security threat exists because the volume of data that the engine must consider is drastically reduced. Also, advantageously, the audit information that is forwarded to the engine is normalized to a standard format, thereby reducing the computational requirements of the engine. The method of reducing the volume of native audit data includes comparing each of the native audits against at least one template and against at least one native audit. By matching the native audits against templates of native audits that do not pose security threats, the native audits that do not pose security threats can be reduced out from further consideration. The native audits that are determined to pose potential security threats are transformed into a standardized format for further analysis by a misuse and intrusion detection engine.

21 Claims, 39 Drawing Sheets

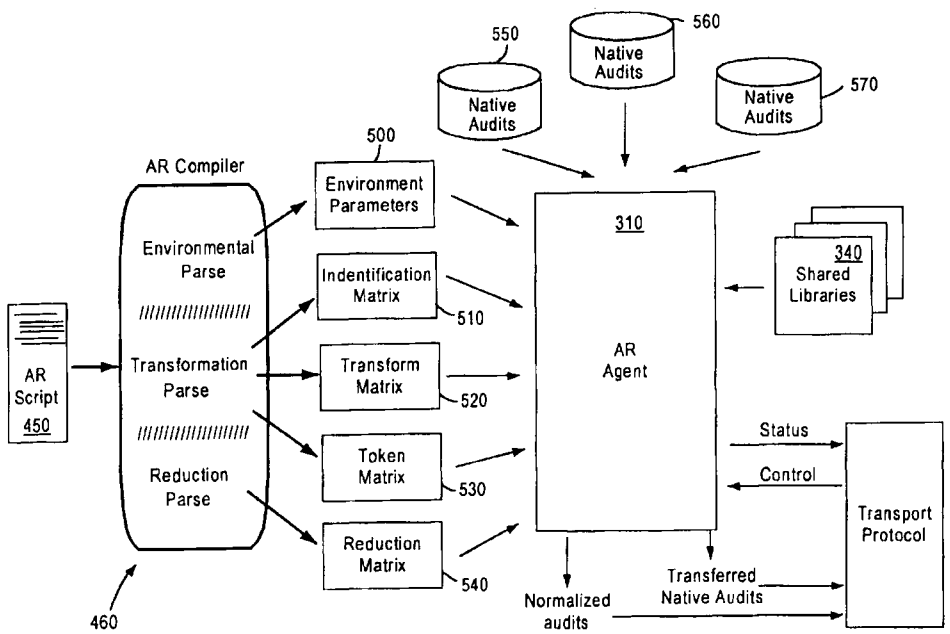
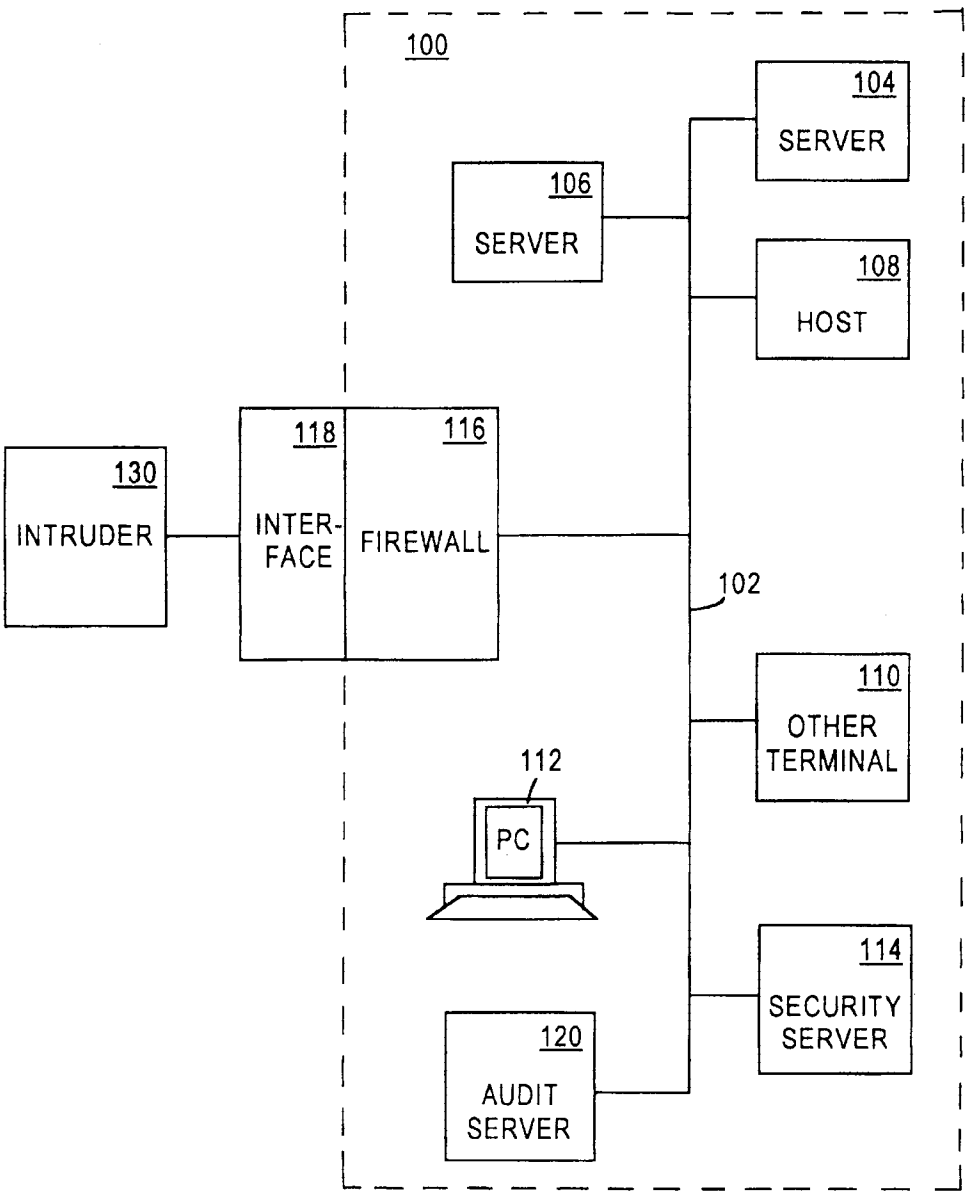


FIG. 1


100
104
SERVER
106
SERVER
108
HOST
118
INTER-
FACE
116
FIREWALL
130
INTRUDER
102
110
OTHER
TERMINAL
112
PC
114
SECURITY
SERVER
120
AUDIT
SERVER

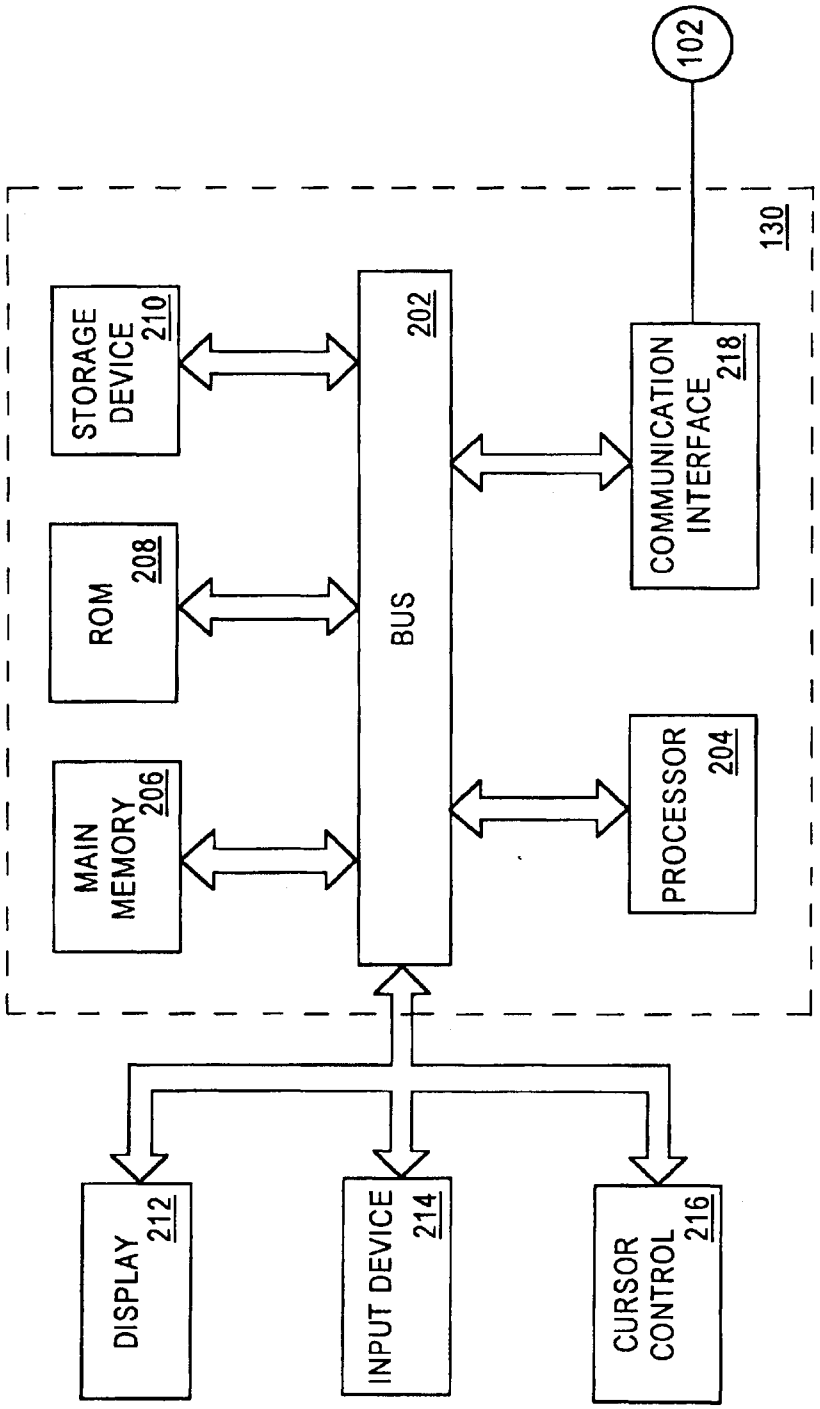

102
130
STORAGE DEVICE 210
ROM 208
MAIN MEMORY 206
BUS 202
COMMUNICATION INTERFACE 218
PROCESSOR 204
DISPLAY 212
INPUT DEVICE 214
CURSOR CONTROL 216

FIG. 2

FIG. 3

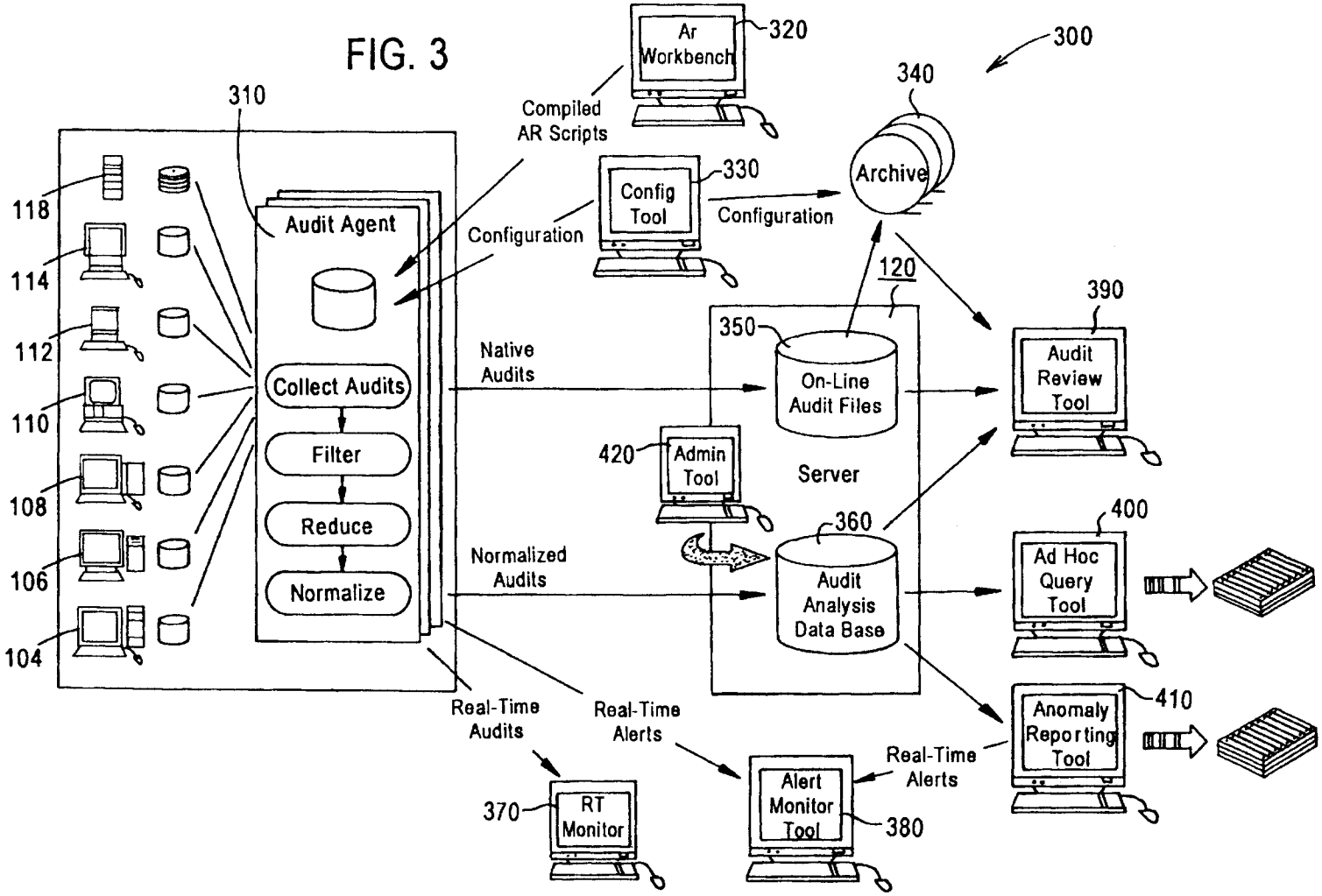
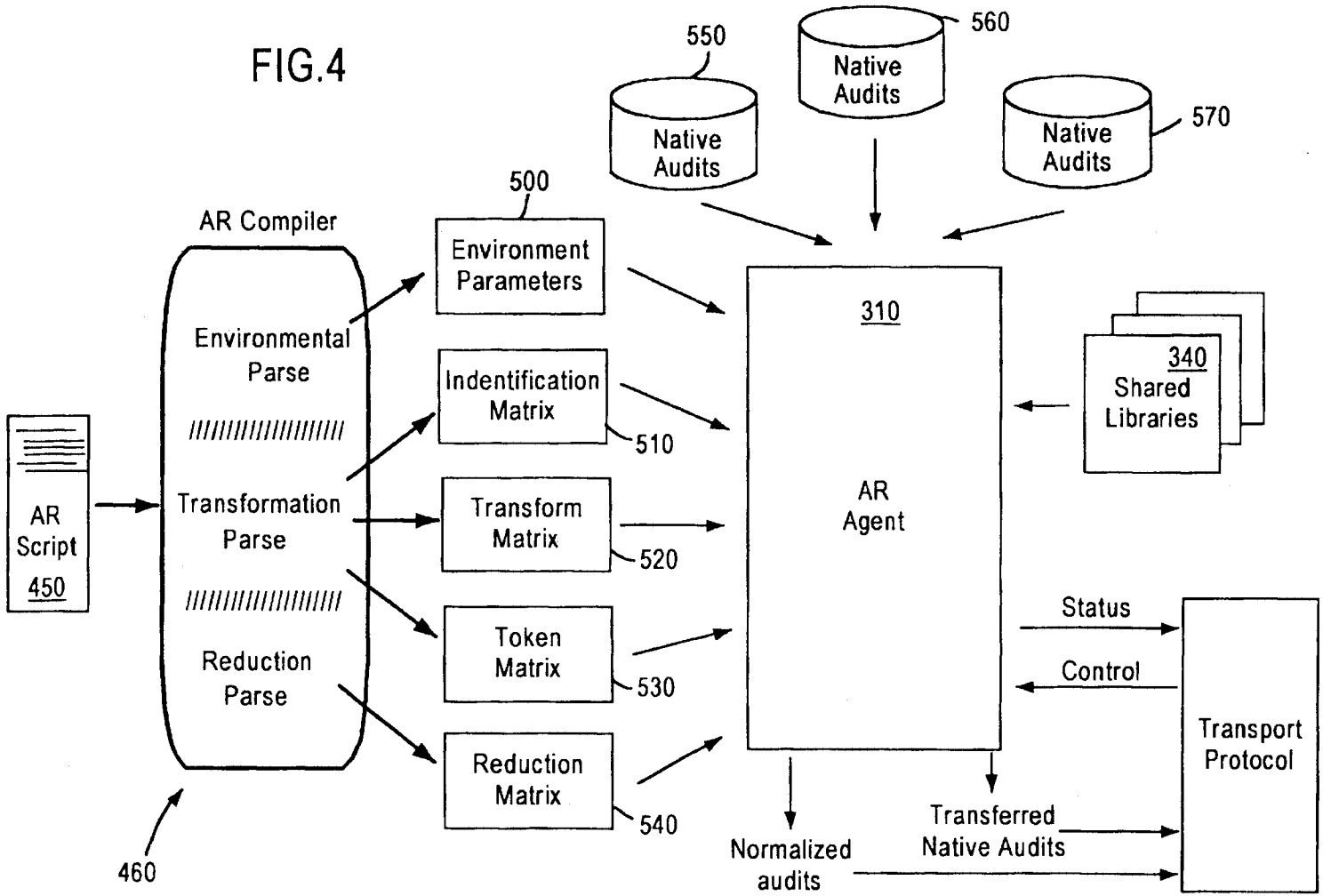

300
310
Ar Workbench
320
Compiled AR Scripts
340
Archive
Config Tool
330
Configuration
Configuration
120
390
Audit Review Tool
350
On-Line Audit Files
Native Audits
Audit Agent
118
114
112
110
108
106
104
Collect Audits
Filter
Reduce
Normalize
420
Admin Tool
Server
360
Audit Analysis Data Base
400
Ad Hoc Query Tool
Normalized Audits
Anomaly Reporting Tool
410
Real-Time Audits
Real-Time Alerts
Real-Time Alerts
370
RT Monitor
Alert Monitor Tool
380

FIG.4


550
560
Native Audits
Native Audits
Native Audits
570
500
AR Compiler
Environment Parameters
Environmental Parse
Indentification Matrix
510
Transformation Parse
Transform Matrix
520
Token Matrix
530
Reduction Parse
Reduction Matrix
540
460
AR Script
450
310
AR Agent
340
Shared Libraries
Status
Control
Transport Protocol
Normalized audits
Transferred Native Audits

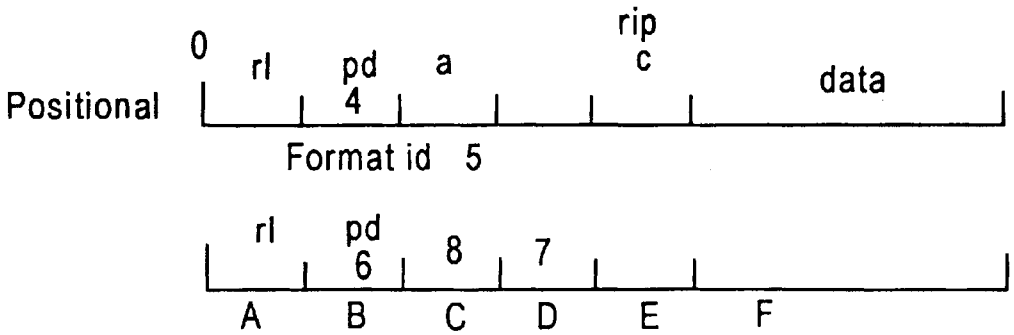

0
rl
pd
4
a
rip
c
data
Positional
Format id 5
rl
pd
6
8
7
A
B
C
D
E
F

FIG. 5A

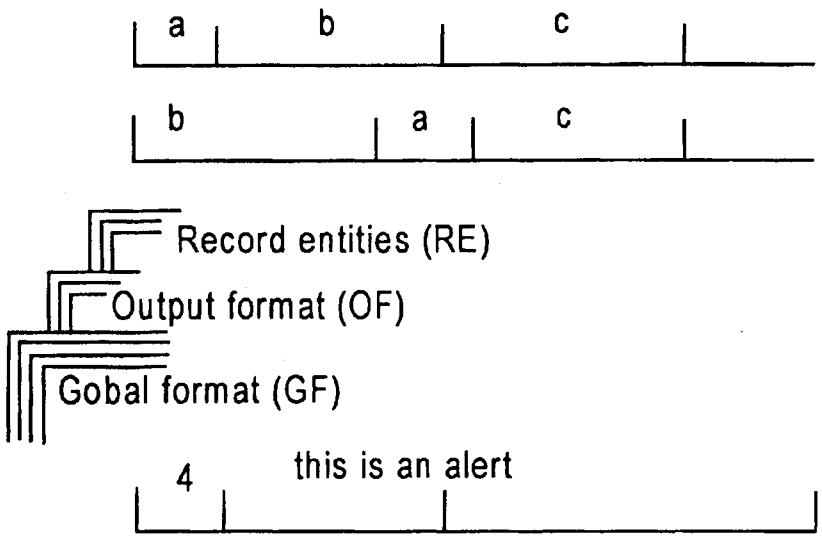

a
b
c
b
a
c
Record entities (RE)
Output format (OF)
Gobal format (GF)
4
this is an alert

FIG. 5B

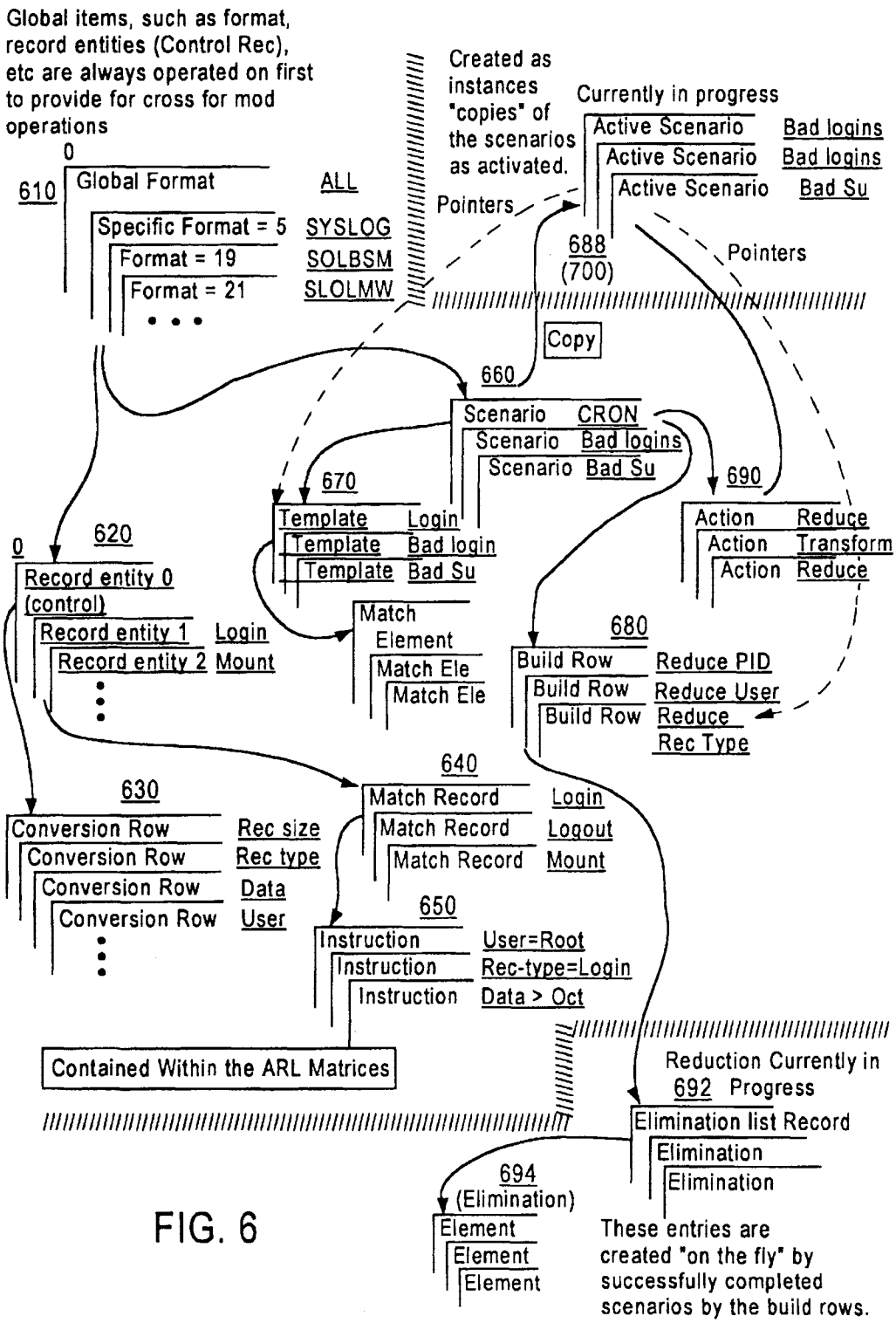

Global items, such as format, record entities (Control Rec), etc are always operated on first to provide for cross for mod operations
0
610
Global Format ALL
Specific Format = 5 SYSLOG
Format = 19 SOLBSM
Format = 21 SLOLMW
Created as instances "copies" of the scenarios as activated.
Currently in progress
Active Scenario Bad logins
Active Scenario Bad logins
Active Scenario Bad Su
Pointers
688
(700)
Pointers
Copy
660
Scenario CRON
Scenario Bad logins
Scenario Bad Su
670
Template Login
Template Bad login
Template Bad Su
690
Action Reduce
Action Transform
Action Reduce
0
620
Record entity 0 (control)
Record entity 1 Login
Record entity 2 Mount
Match Element
Match Ele
Match Ele
680
Build Row Reduce PID
Build Row Reduce User
Build Row Reduce Rec Type
640
630
Match Record Login
Match Record Logout
Match Record Mount
Conversion Row Rec size
Conversion Row Rec type
Conversion Row Data
Conversion Row User
650
Instruction User=Root
Instruction Rec-type=Login
Instruction Data > Oct
Contained Within the ARL Matrices
Reduction Currently in
692 Progress
Elimination list Record
Elimination
Elimination
694
(Elimination)
Element
Element
Element
These entries are created "on the fly" by successfully completed scenarios by the build rows.

FIG. 6

FIG. 7A

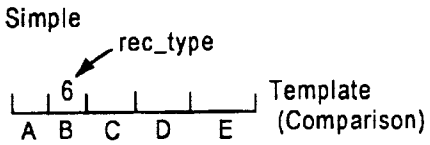


FIG. 7B

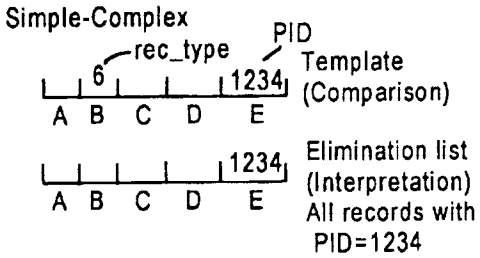


FIG. 7C

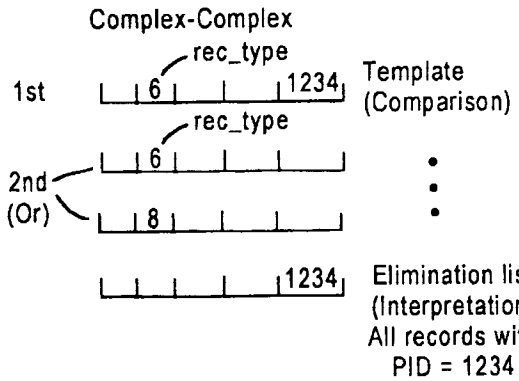


FIG. 7D

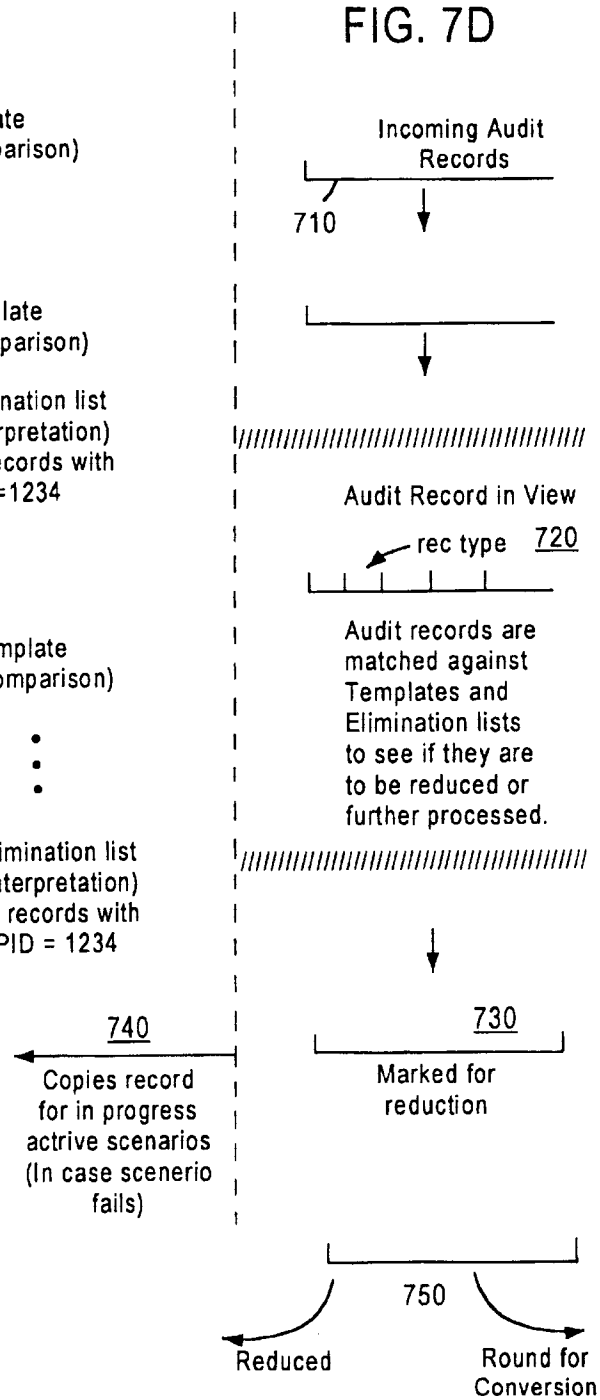


Fig. 8

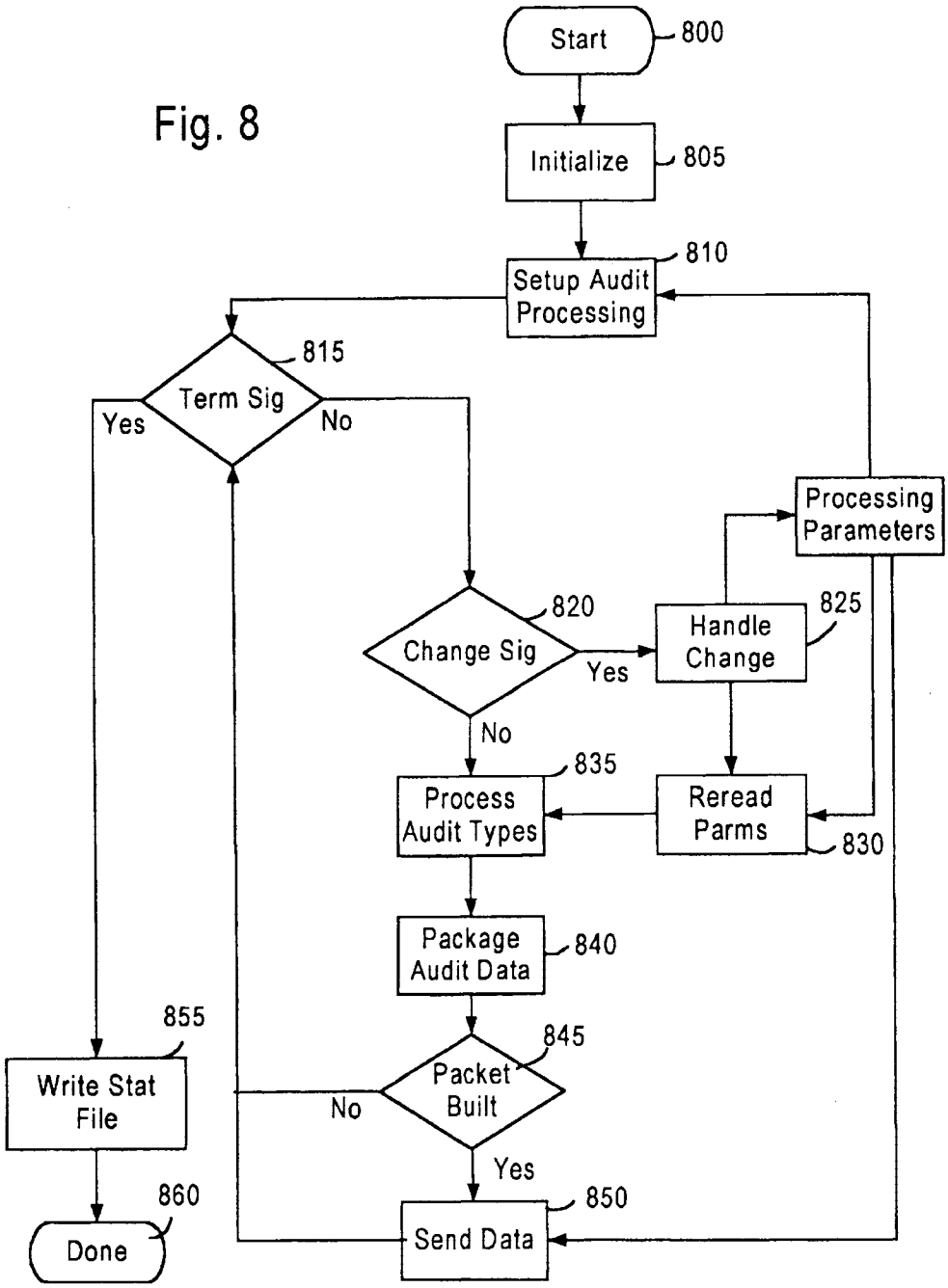

Start
800
Initialize
805
Setup Audit Processing
810
Term Sig
815
Yes
No
Processing Parameters
Change Sig
820
Yes
Handle Change
825
No
Process Audit Types
835
Reread Parms
830
Package Audit Data
840
Packet Built
845
No
Yes
Write Stat File
855
Done
860
Send Data
850

FIG. 9

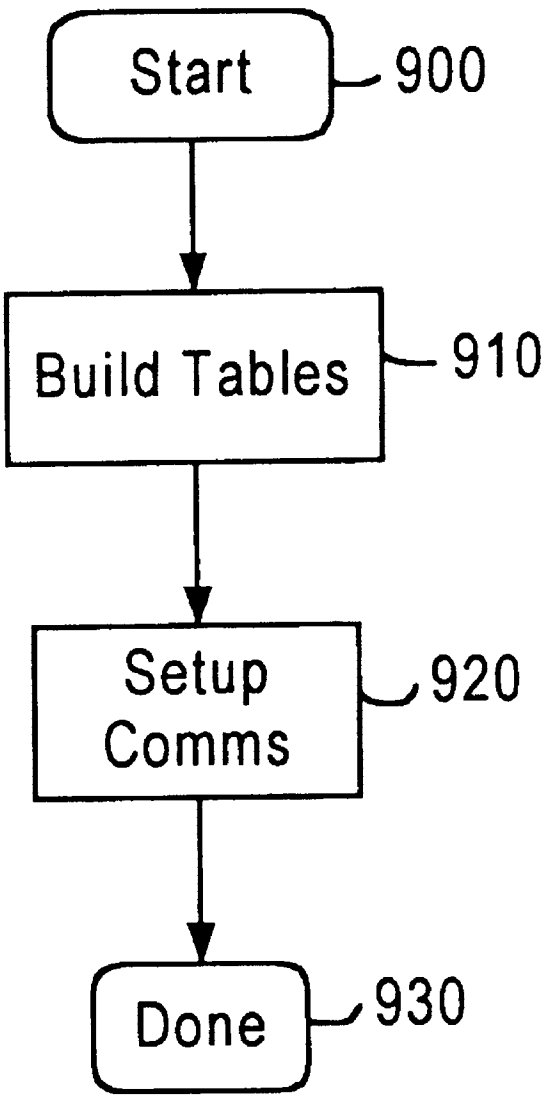


FIG. 10

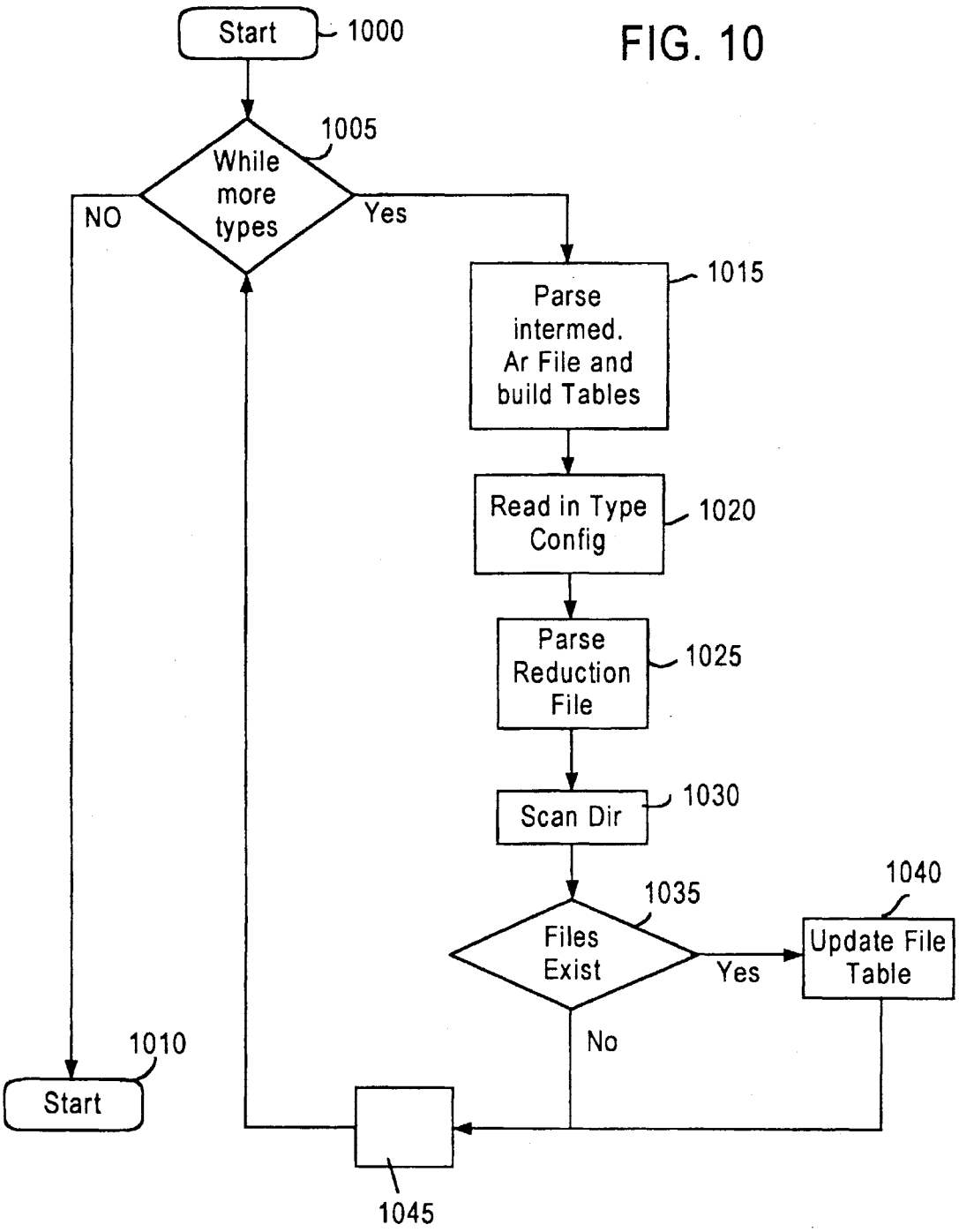
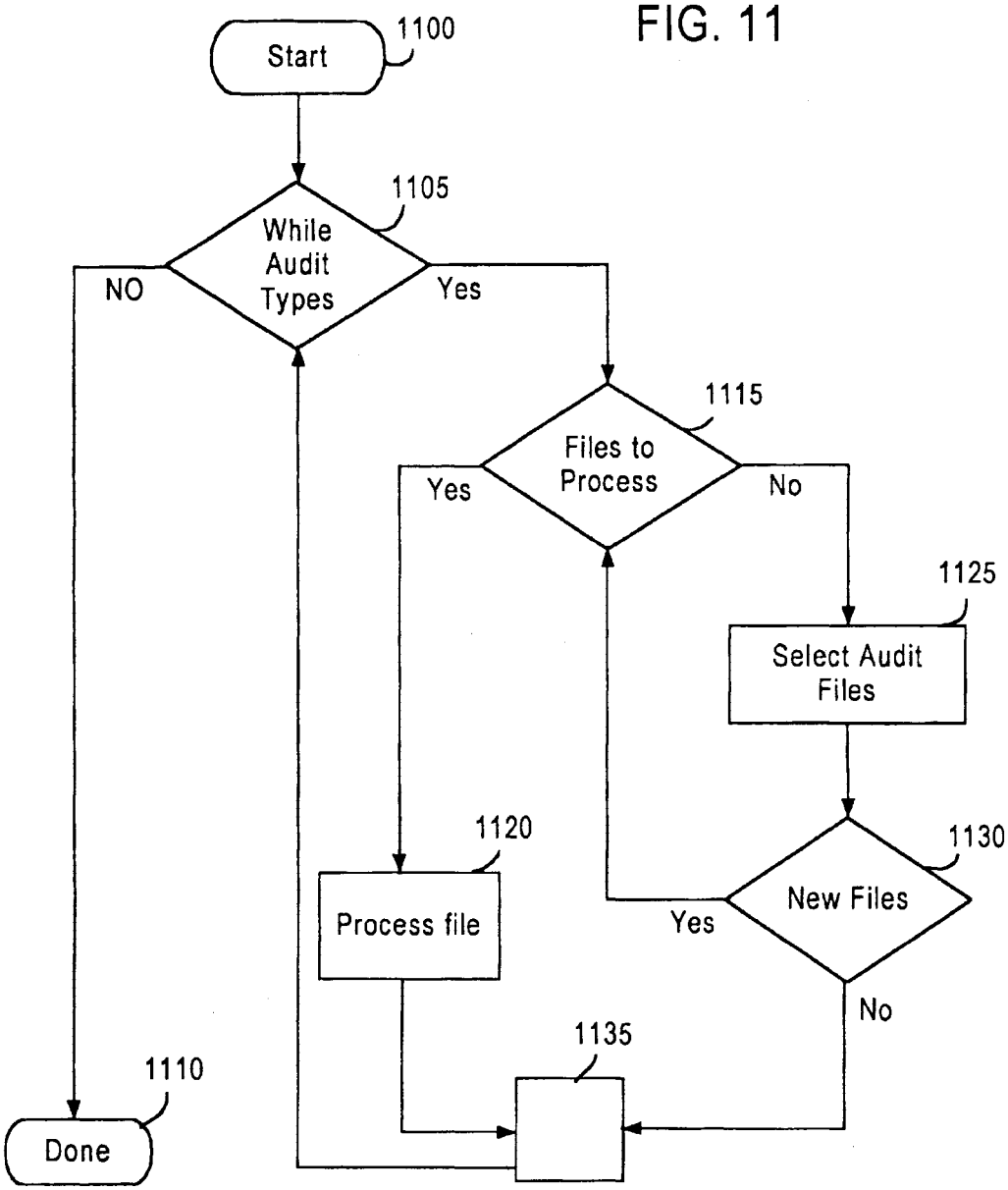

Start
1000
While more types
1005
NO
Yes
Parse intermed. Ar File and build Tables
1015
Read in Type Config
1020
Parse Reduction File
1025
Scan Dir
1030
Files Exist
1035
Yes
Update File Table
1040
No
Start
1010
1045

FIG. 11


Start
1100
While Audit Types
1105
NO
Yes
Files to Process
1115
Yes
No
Select Audit Files
1125
Process file
1120
New Files
1130
Yes
No
1135
Done
1110

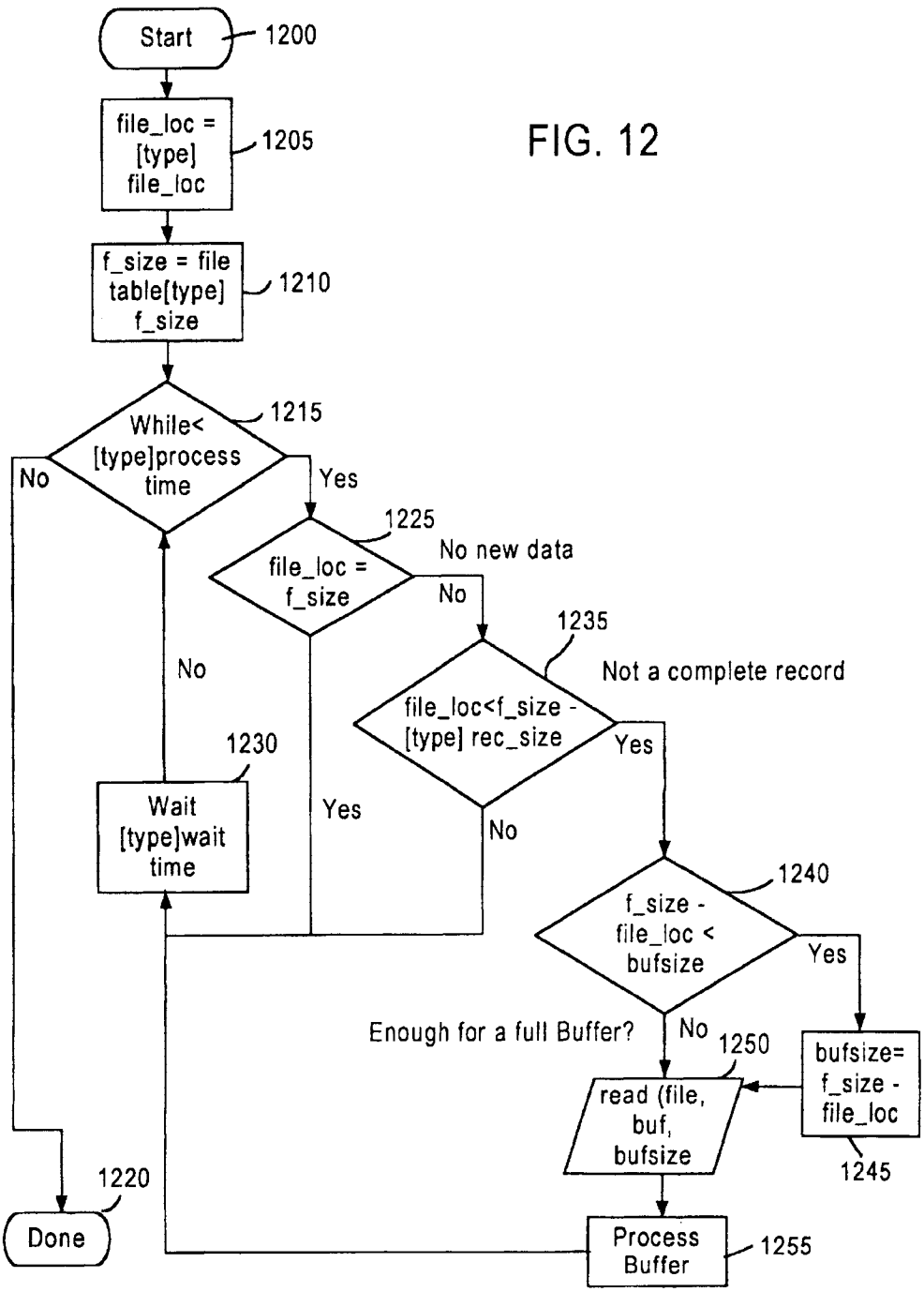

Start
1200
file_loc = [type] file_loc
1205
f_size = file table[type] f_size
1210
While< [type]process time
1215
No
Yes
1225
file_loc = f_size
No new data
No
1235
Not a complete record
file_loc<f_size - [type] rec_size
Yes
No
1230
Wait [type]wait time
Yes
No
1240
f_size - file_loc < bufsize
Yes
Enough for a full Buffer?
No
1250
bufsize= f_size - file_loc
1245
read (file, buf, bufsize
1220
Done
Process Buffer
1255

FIG. 12

FIG. 13

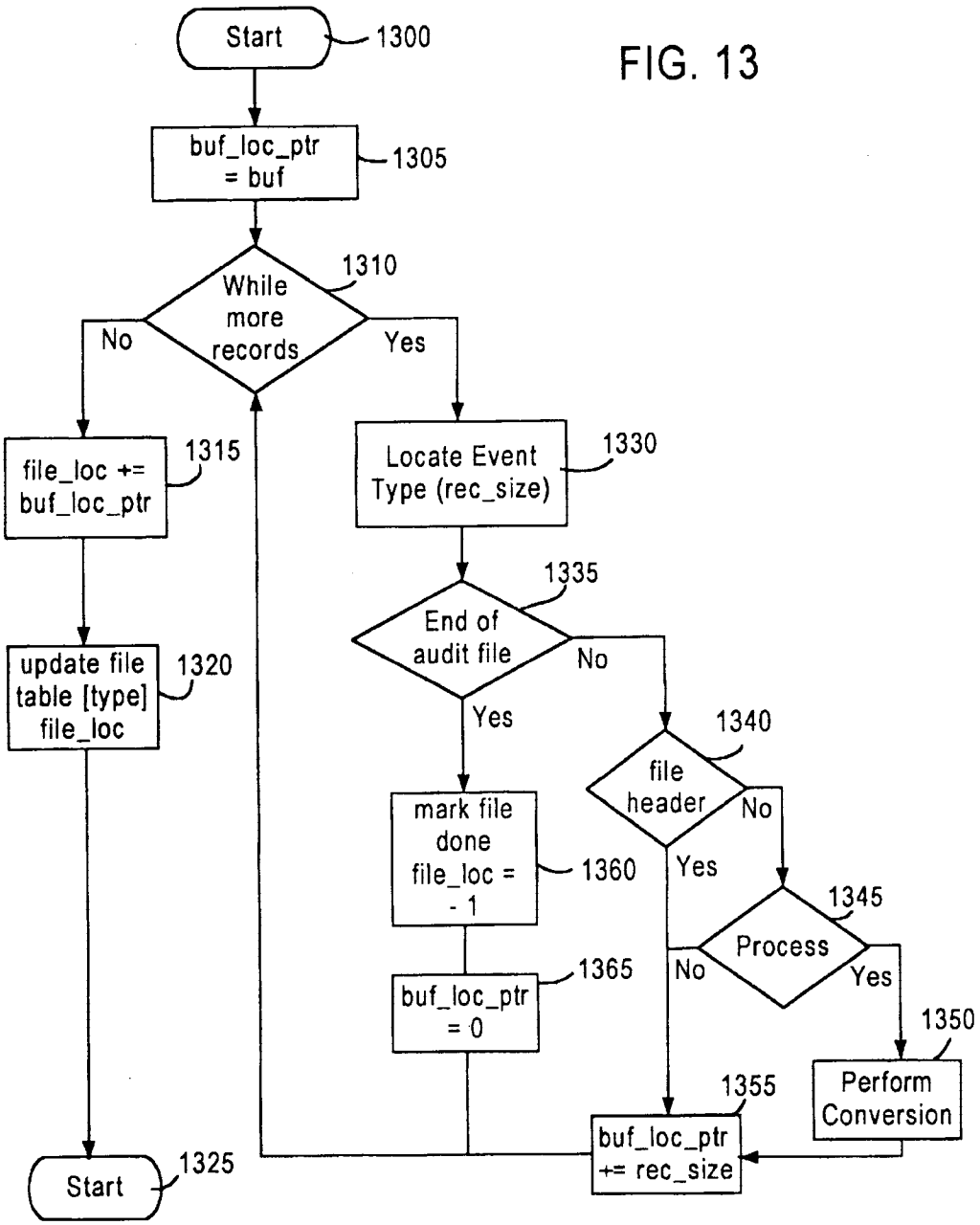

Start
1300
buf_loc_ptr = buf
1305
While more records
1310
No
Yes
file_loc += buf_loc_ptr
1315
update file table [type] file_loc
1320
Start
1325
Locate Event Type (rec_size)
1330
End of audit file
1335
No
Yes
mark file done file_loc = - 1
1360
buf_loc_ptr = 0
1365
file header
1340
No
Yes
Process
1345
No
Yes
Perform Conversion
1350
buf_loc_ptr += rec_size
1355

FIG. 14

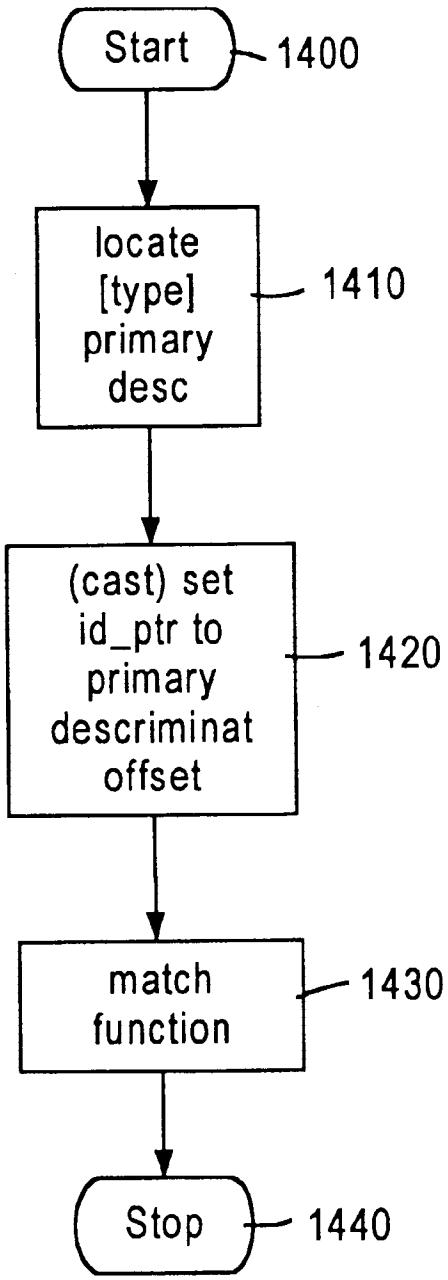

Start
1400
locate [type] primary desc
1410
(cast) set id_ptr to primary descriminat offset
1420
match function
1430
Stop
1440

FIG. 15

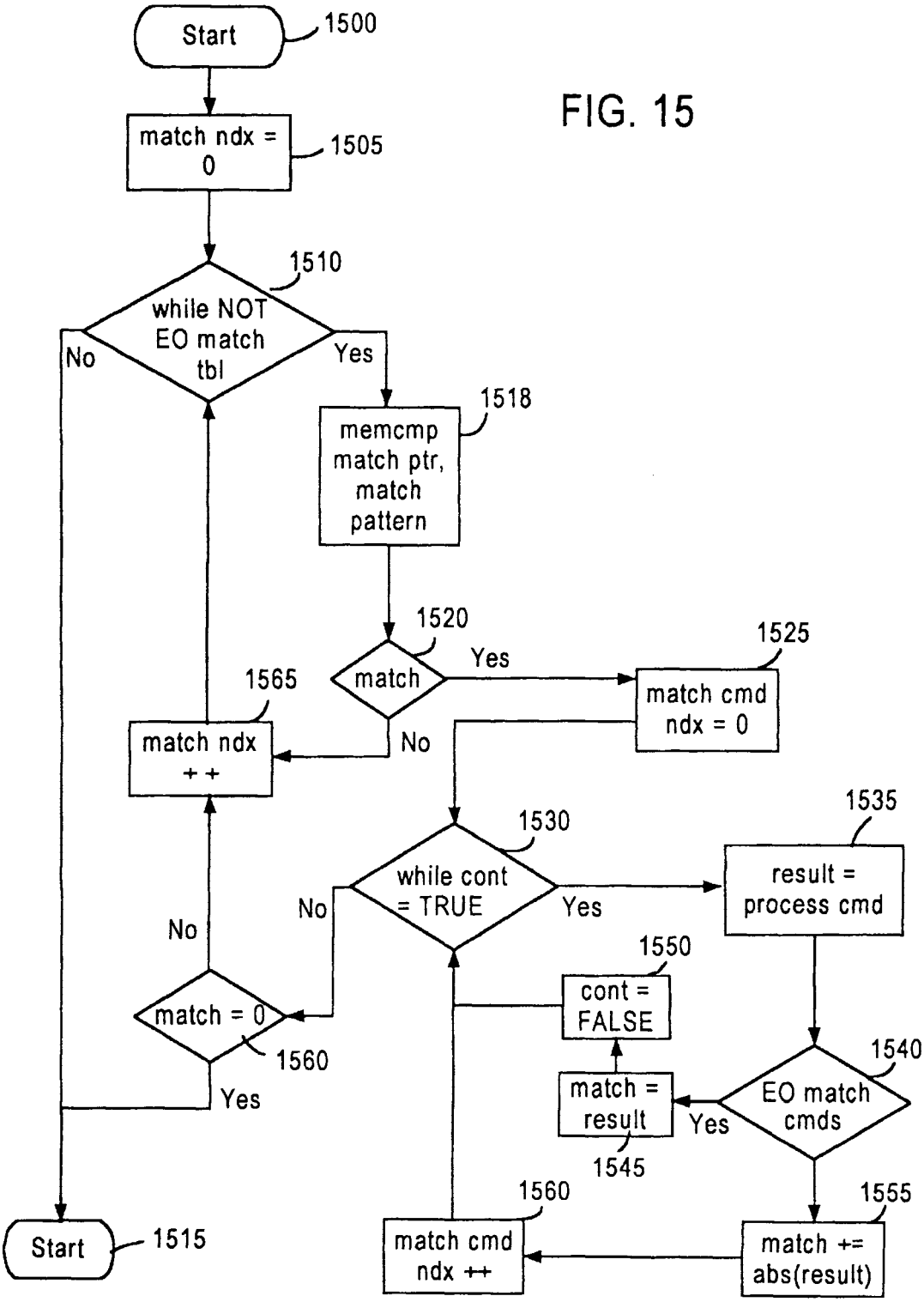

Start
1500
match ndx = 0
1505
1510
while NOT EO match tbl
No
Yes
1518
memcmp match ptr, match pattern
1520
match
Yes
1525
match cmd ndx = 0
1565
match ndx + +
No
1530
while cont = TRUE
No
Yes
1535
result = process cmd
No
1550
cont = FALSE
match = 0
1560
Yes
match = result
1540
EO match cmds
Yes
1545
1560
match cmd ndx ++
1555
match += abs(result)
Start
1515

FIG. 16

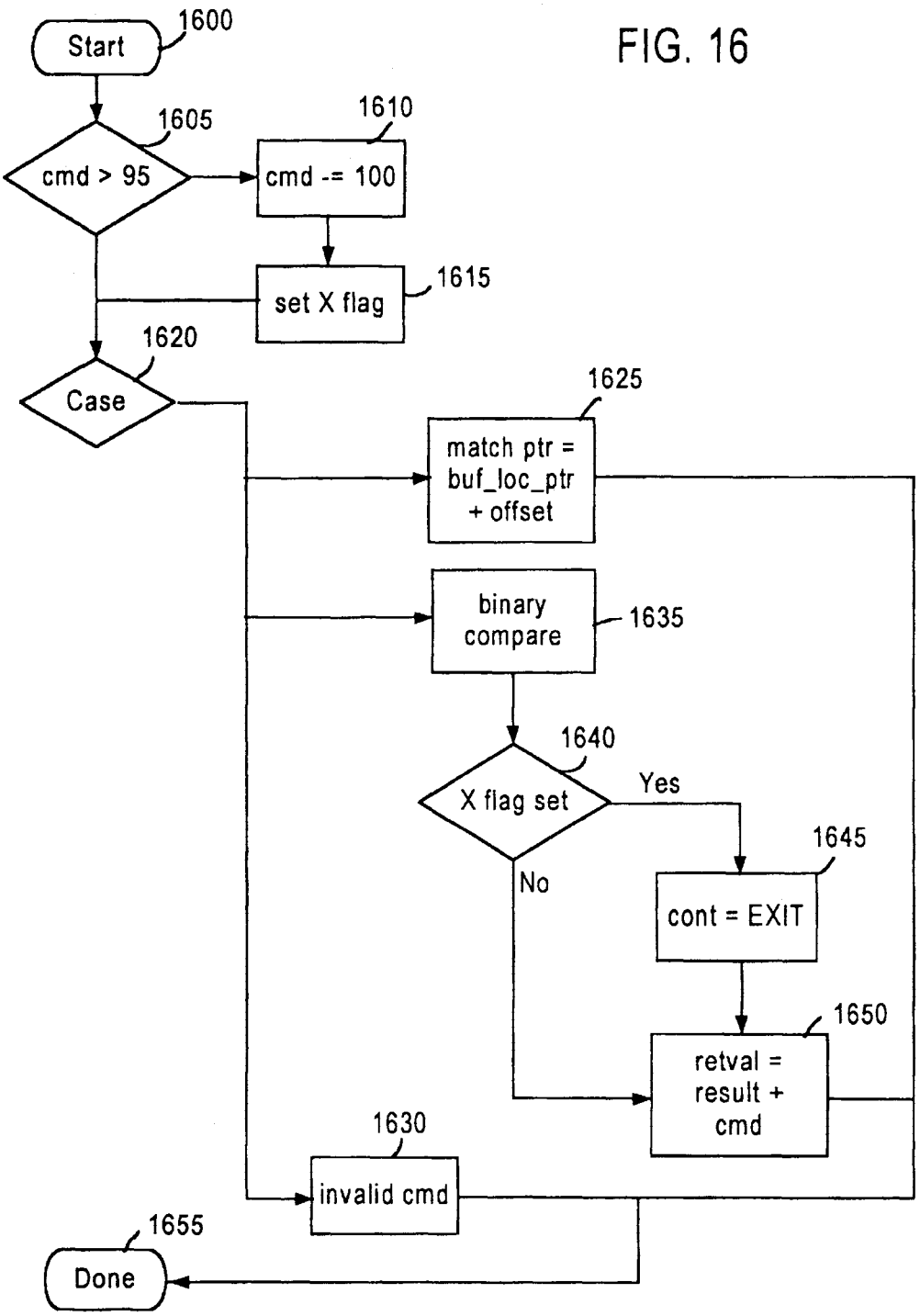

Start
1600
1605
cmd > 95
1610
cmd -= 100
set X flag
1615
1620
Case
1625
match ptr = buf_loc_ptr + offset
binary compare
1635
1640
X flag set
Yes
No
1645
cont = EXIT
1650
retval = result + cmd
1630
invalid cmd
1655
Done

FIG.17

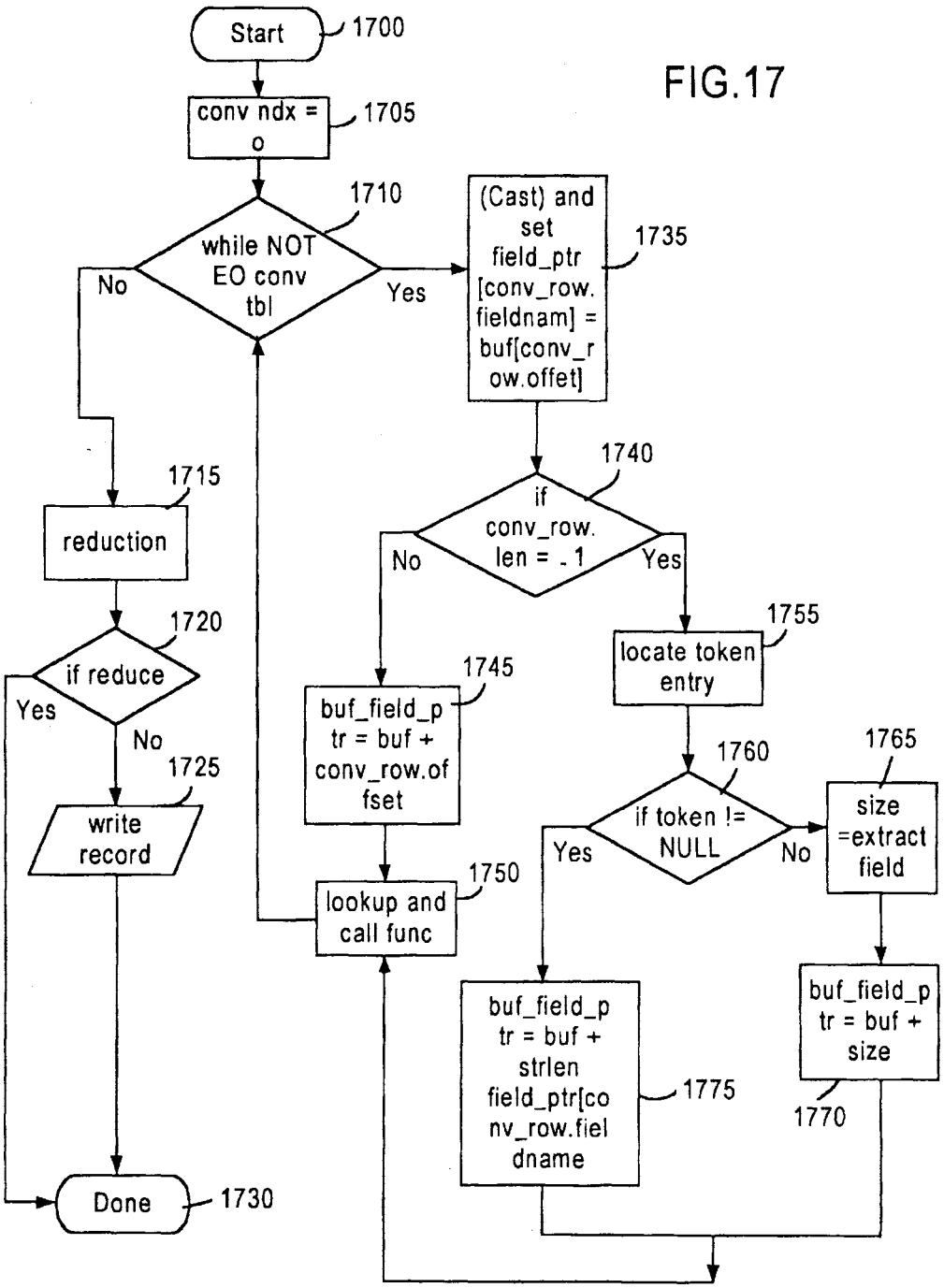

Start
1700
conv ndx = o
1705
1710
while NOT EO conv tbl
No
Yes
(Cast) and set field_ptr [conv_row. fieldnam] = buf[conv_r ow.offet]
1735
1740
if conv_row. len = -1
No
Yes
1715
reduction
1720
if reduce
Yes
No
1725
write record
1745
buf_field_p tr = buf + conv_row.of fset
1755
locate token entry
1760
if token != NULL
Yes
No
1765
size =extract field
1750
lookup and call func
buf_field_p tr = buf + strlen field_ptr[co nv_row.fiel dname
1775
buf_field_p tr = buf + size
1770
Done
1730

FIG. 18

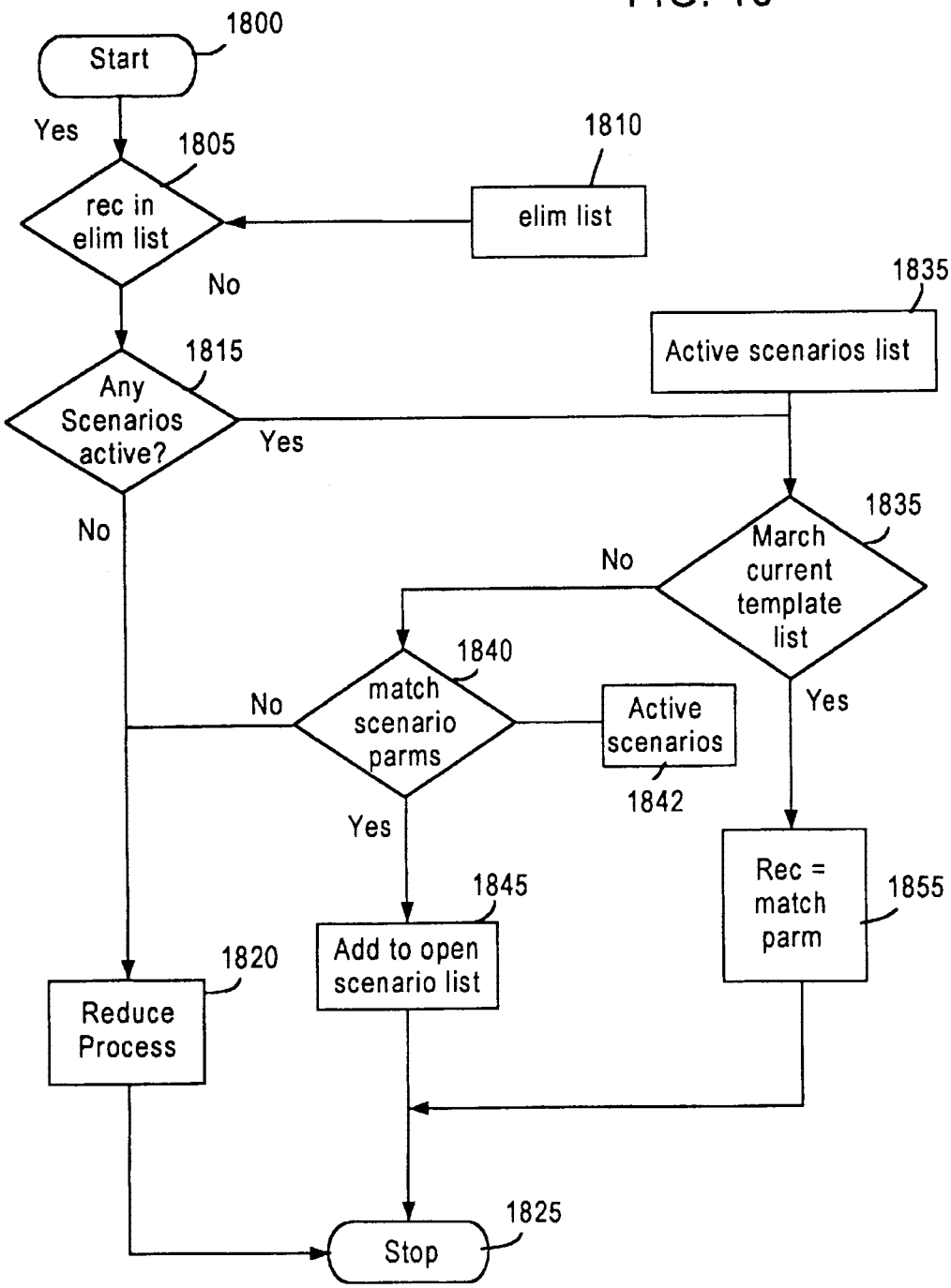

1800
Start
Yes
1805
rec in elim list
1810
elim list
No
1835
Active scenarios list
1815
Any Scenarios active?
Yes
No
1835
March current template list
No
Yes
1840
match scenario parms
No
Active scenarios
1842
Yes
1845
Add to open scenario list
Rec = match parm
1855
1820
Reduce Process
1825
Stop

FIG. 19

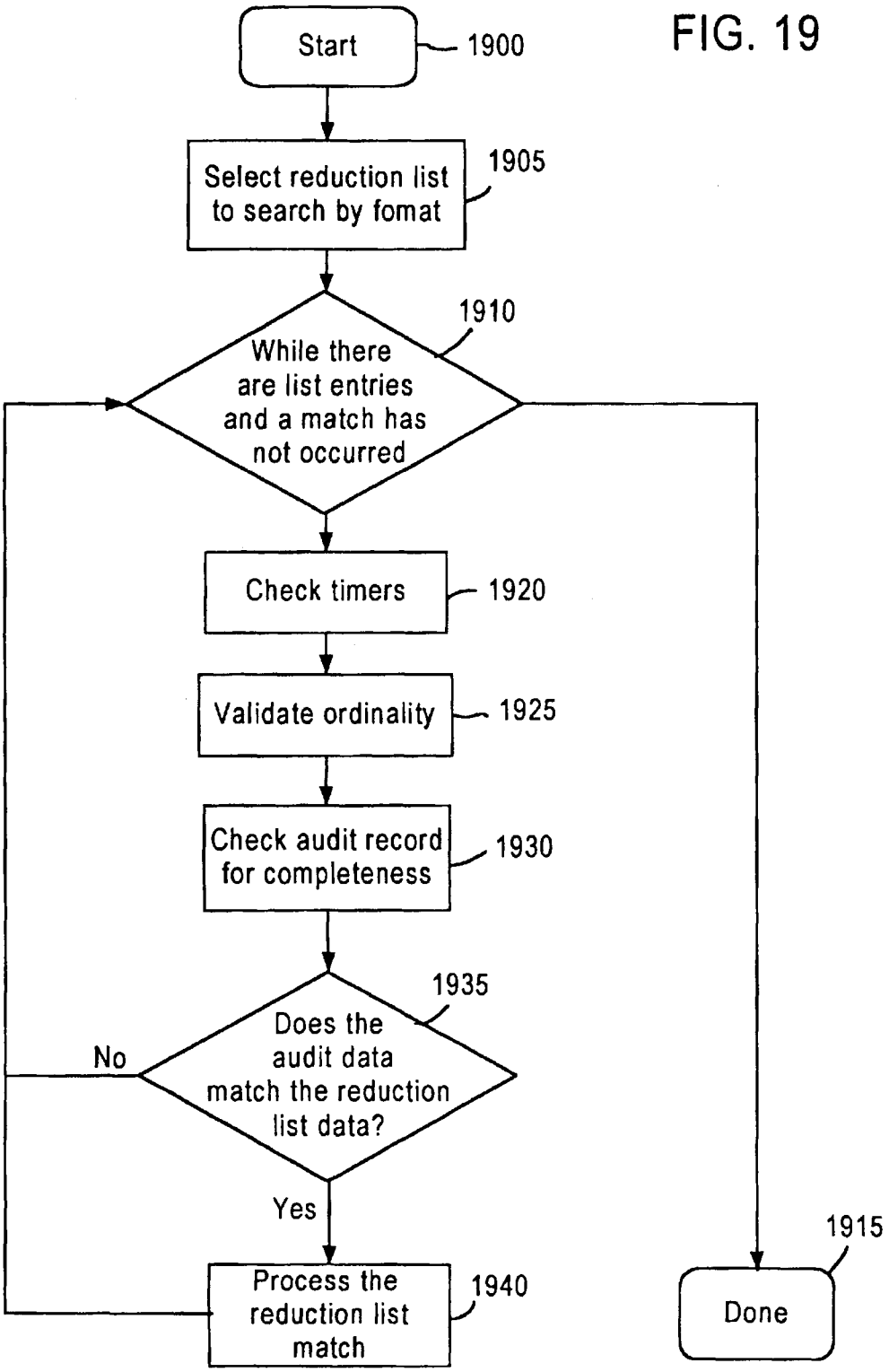

Start
1900
Select reduction list to search by fomat
1905
While there are list entries and a match has not occurred
1910
Check timers
1920
Validate ordinality
1925
Check audit record for completeness
1930
Does the audit data match the reduction list data?
1935
No
Yes
Process the reduction list match
1940
Done
1915

FIG.20

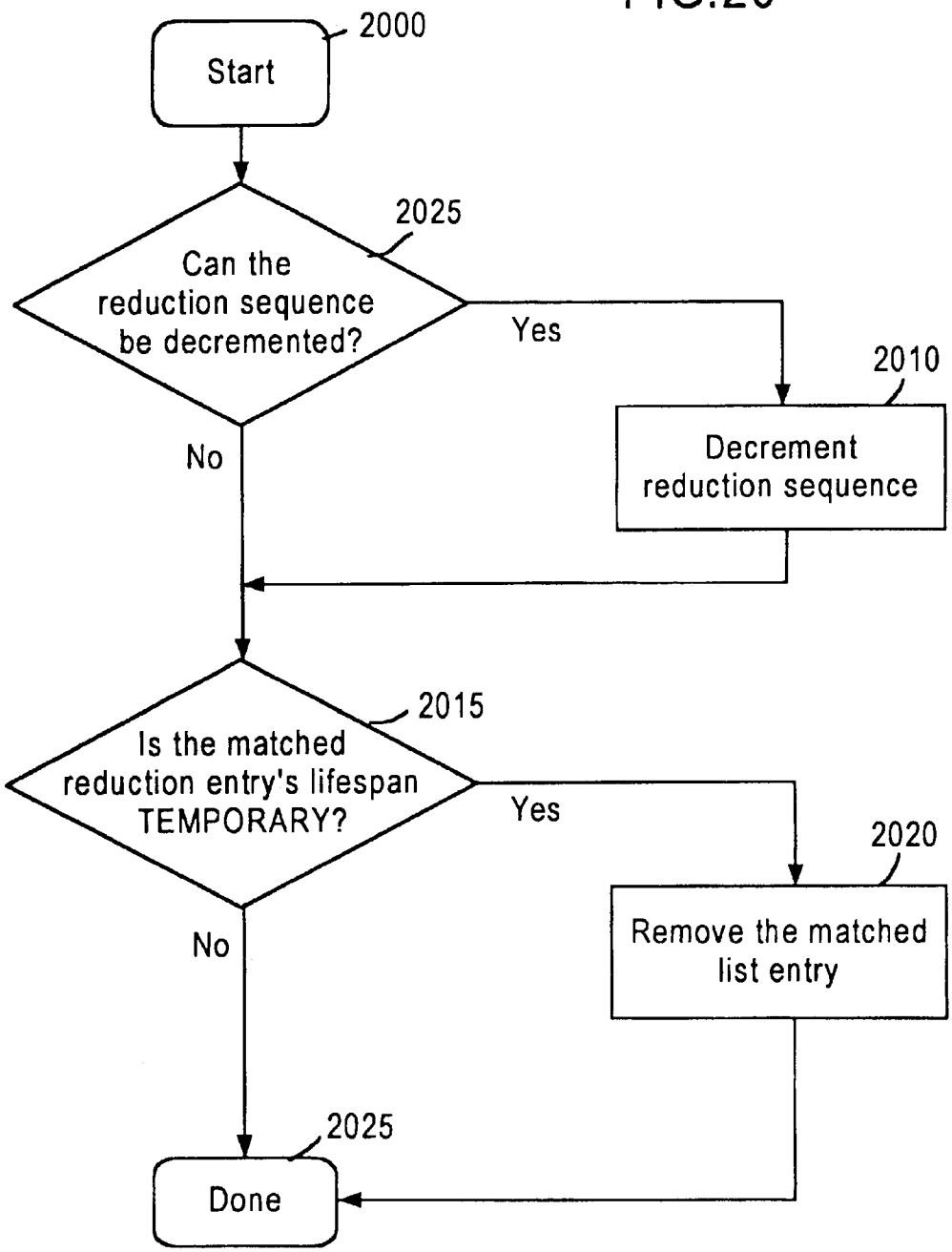

2000
Start
2025
Can the reduction sequence be decremented?
Yes
No
2010
Decrement reduction sequence
2015
Is the matched reduction entry's lifespan TEMPORARY?
Yes
No
2020
Remove the matched list entry
2025
Done

FIG. 21

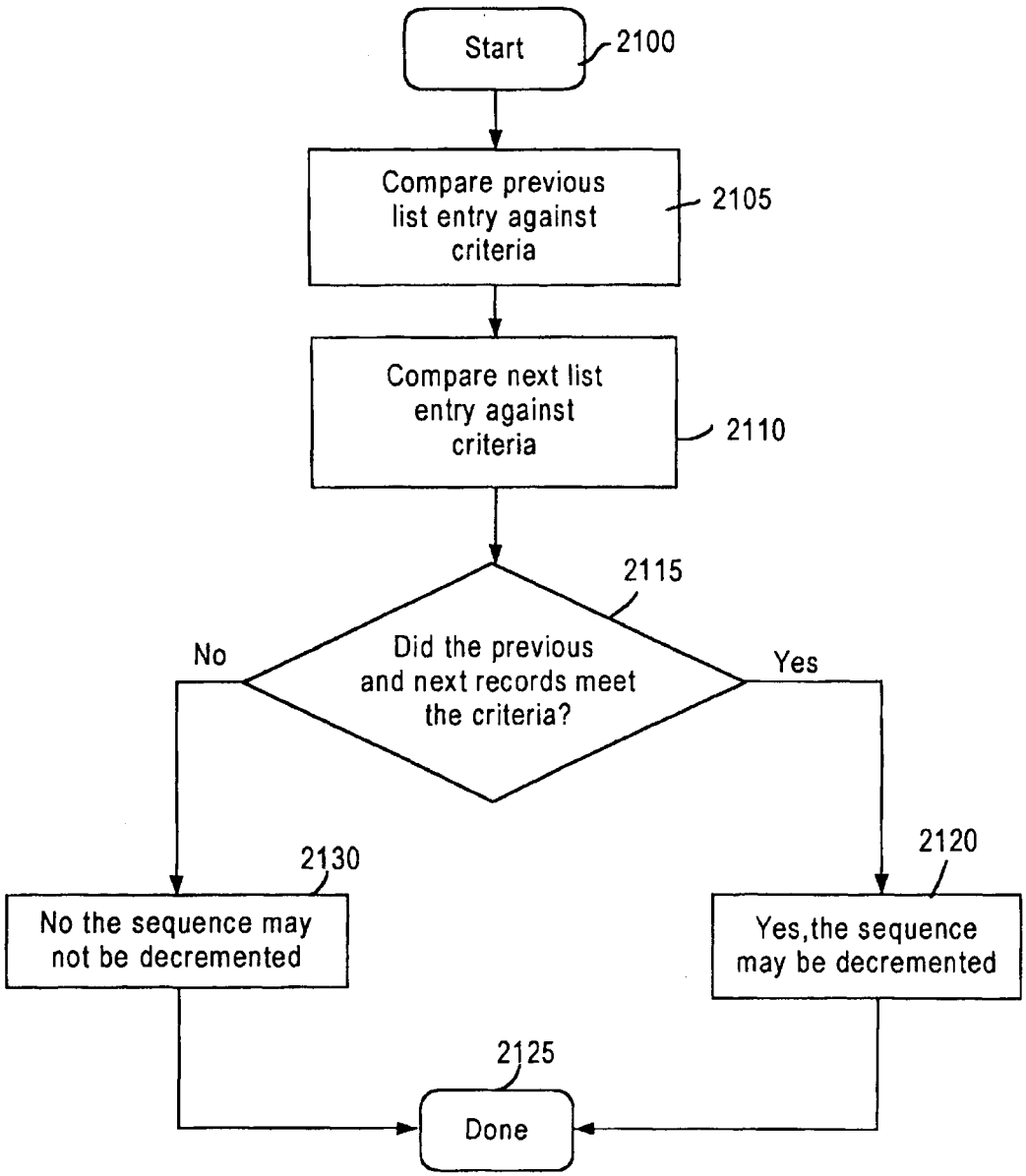

Start
2100
Compare previous list entry against criteria
2105
Compare next list entry against criteria
2110
2115
Did the previous and next records meet the criteria?
No
Yes
2130
No the sequence may not be decremented
2120
Yes,the sequence may be decremented
2125
Done

FIG. 22

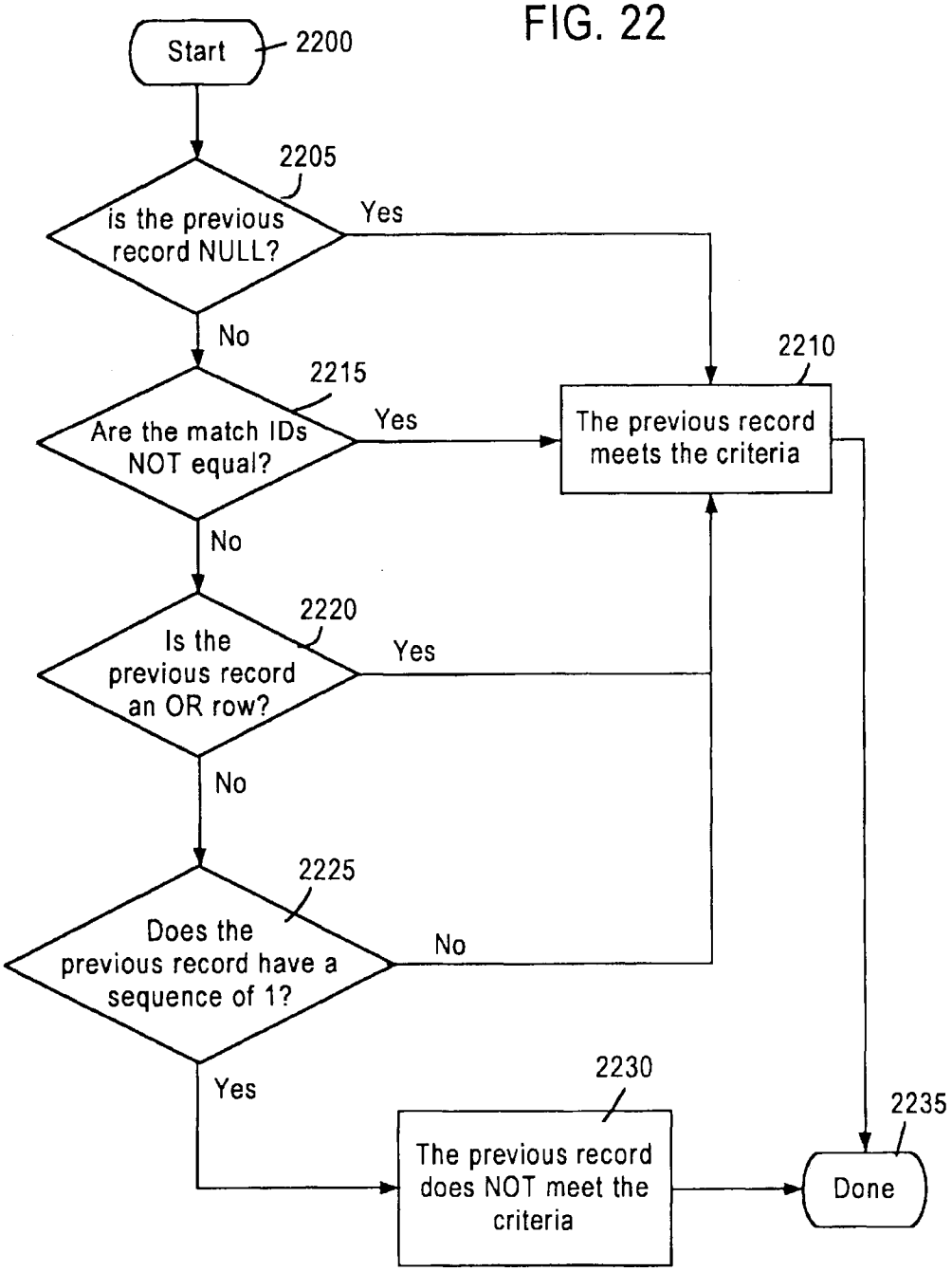

Start
2200
2205
is the previous record NULL?
Yes
No
2210
2215
Are the match IDs NOT equal?
Yes
The previous record meets the criteria
No
2220
Is the previous record an OR row?
Yes
No
2225
Does the previous record have a sequence of 1?
No
Yes
2230
The previous record does NOT meet the criteria
2235
Done

FIG. 23

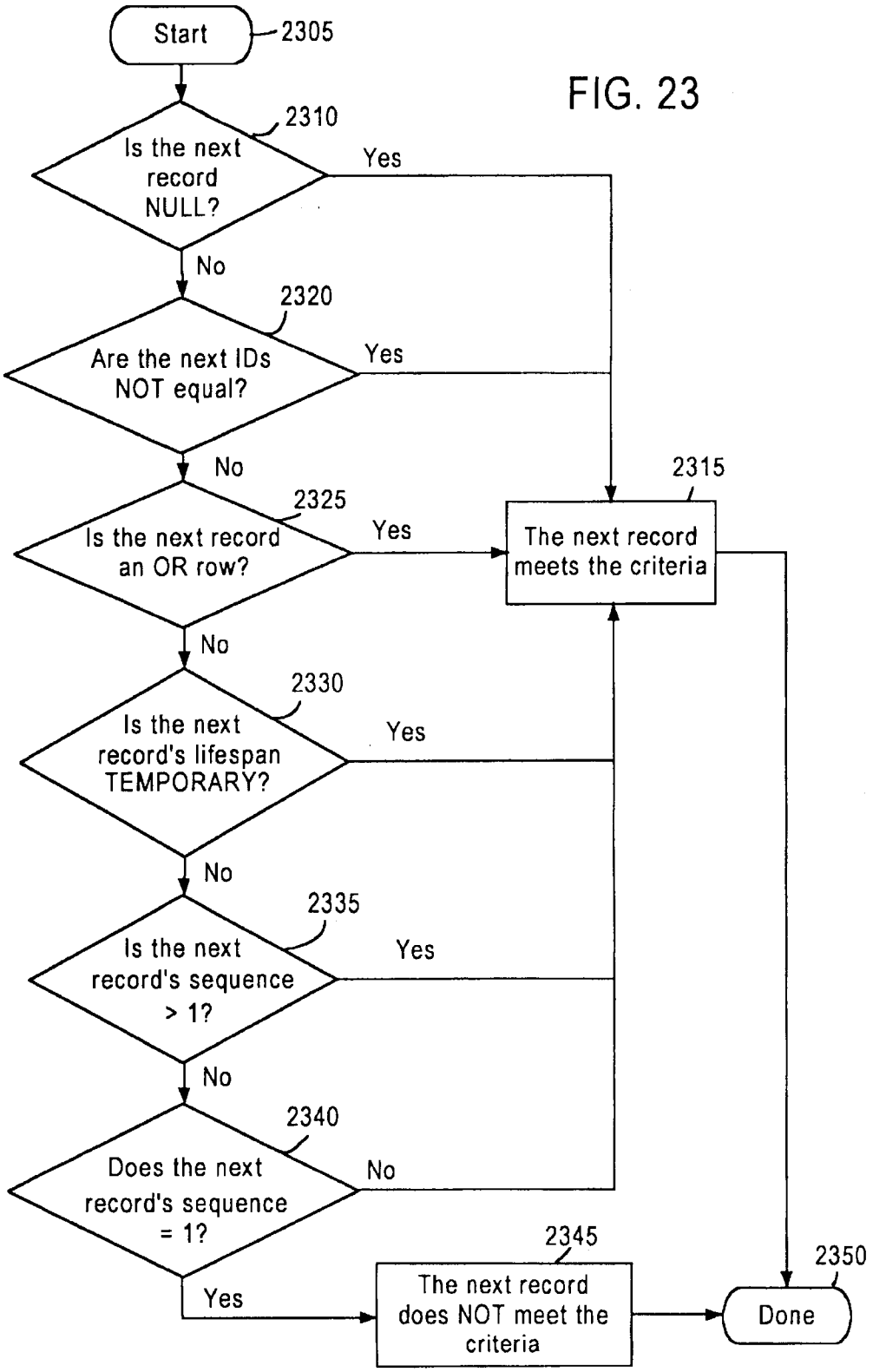

Start
2305
Is the next record NULL?
2310
Yes
No
Are the next IDs NOT equal?
2320
Yes
No
Is the next record an OR row?
2325
Yes
No
Is the next record's lifespan TEMPORARY?
2330
Yes
No
Is the next record's sequence > 1?
2335
Yes
No
Does the next record's sequence = 1?
2340
No
Yes
The next record meets the criteria
2315
The next record does NOT meet the criteria
2345
Done
2350

FIG. 24

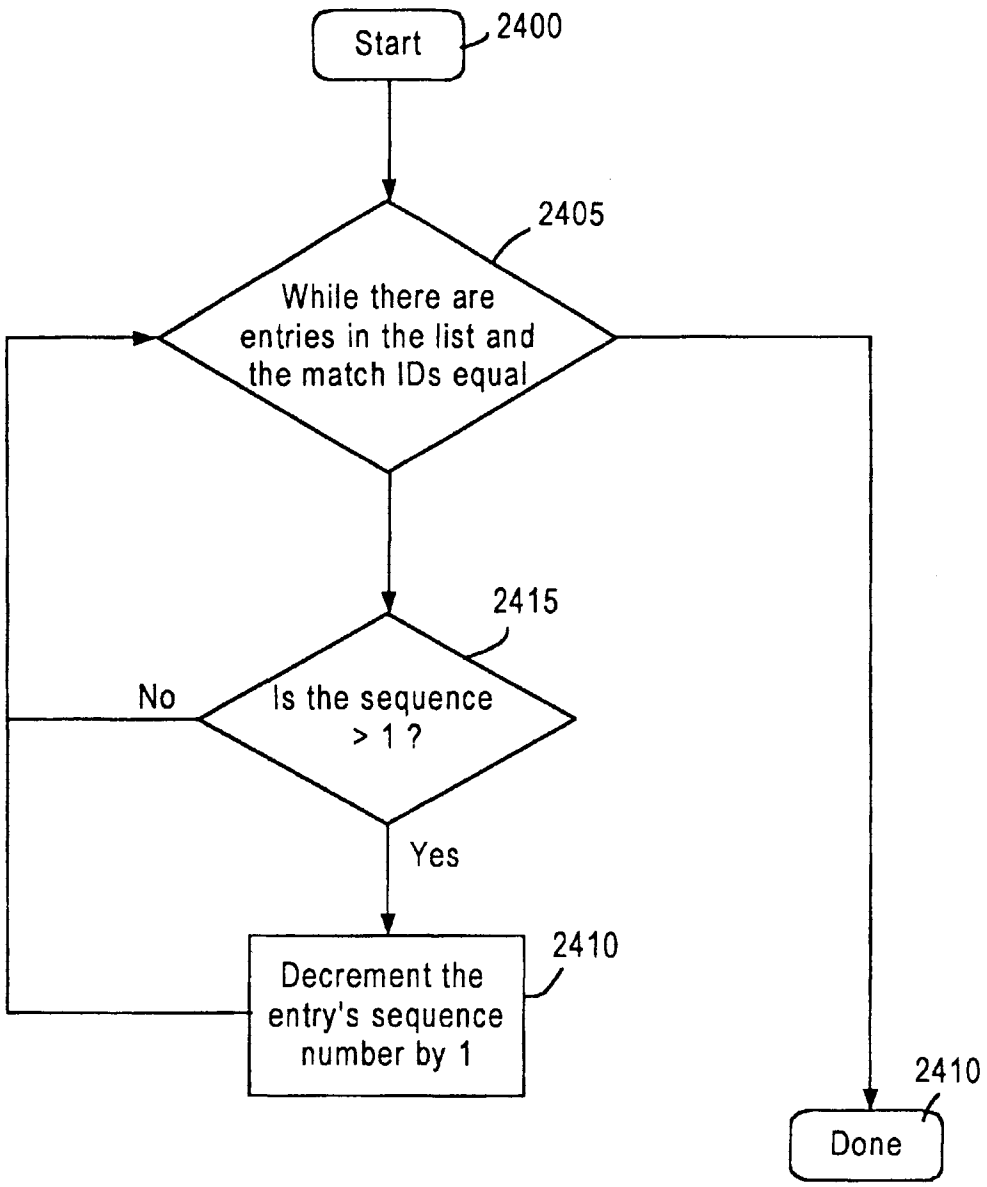

Start
2400
2405
While there are entries in the list and the match IDs equal
2415
Is the sequence > 1 ?
No
Yes
2410
Decrement the entry's sequence number by 1
2410
Done

FIG. 25

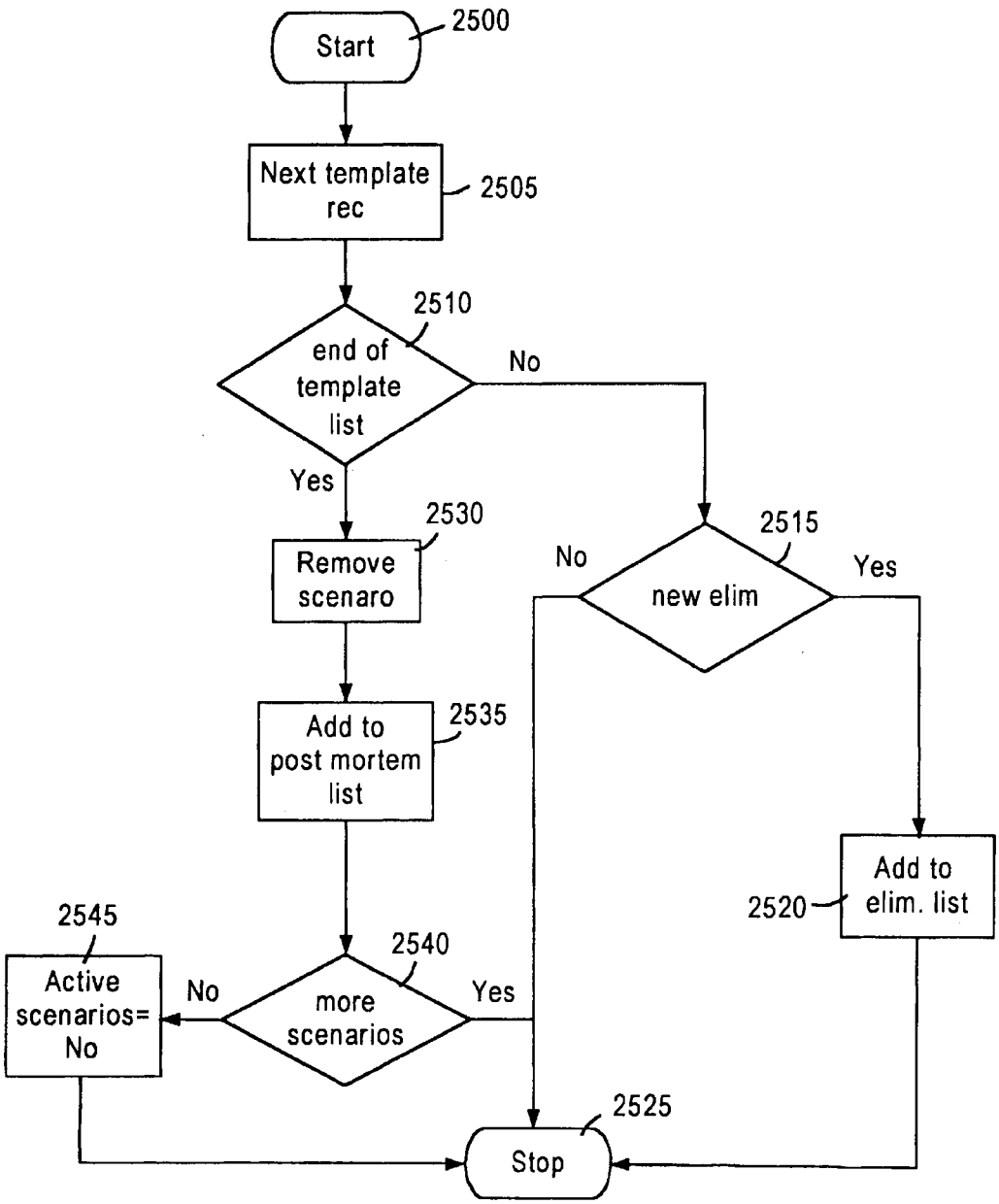

Start
2500
Next template rec
2505
2510
end of template list
No
Yes
2530
Remove scenaro
2515
No
new elim
Yes
2535
Add to post mortem list
Add to elim. list
2520
2545
Active scenarios= No
2540
No
more scenarios
Yes
2525
Stop

FIG. 26

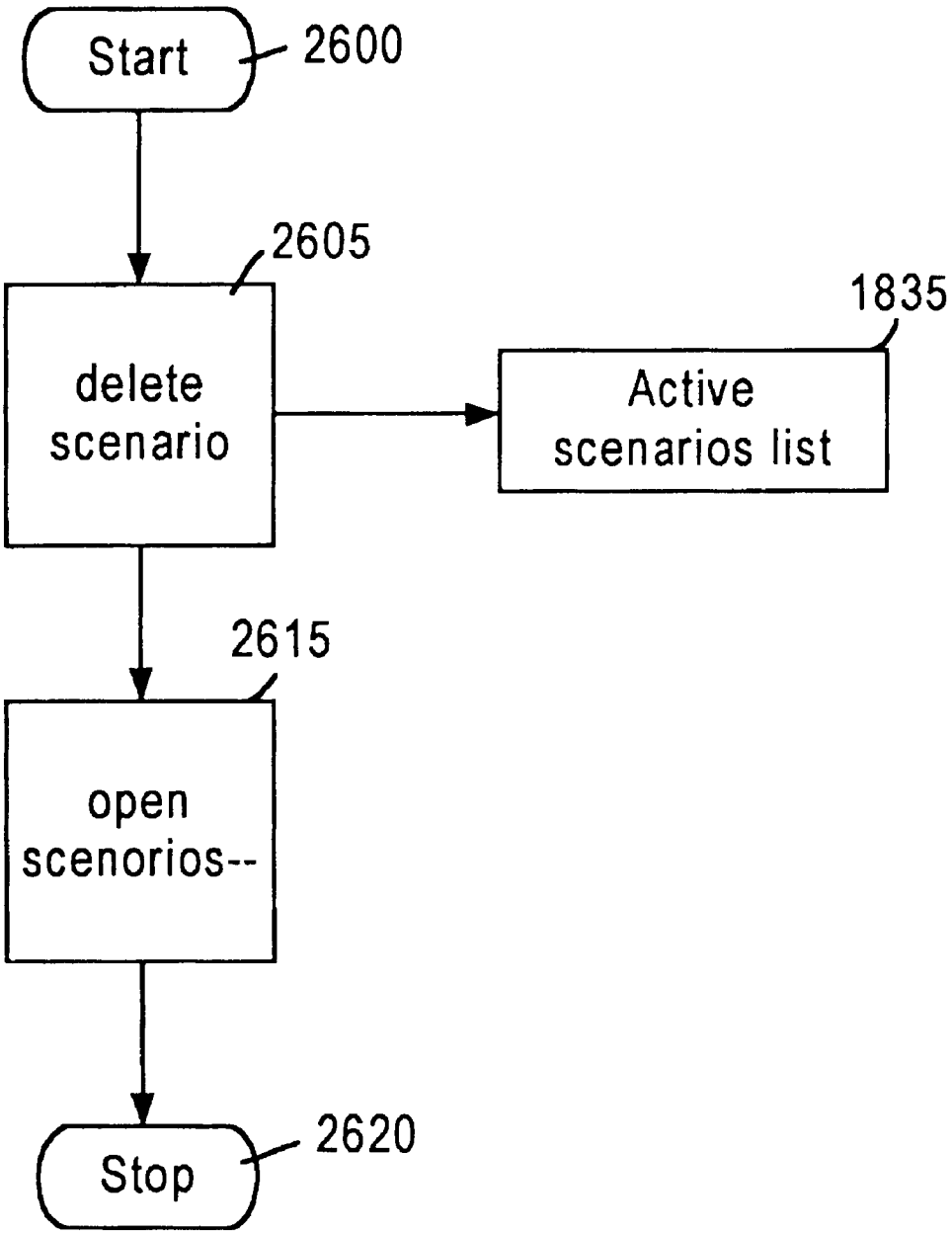

Start
2600
2605
delete scenario
1835
Active scenarios list
2615
open scenorios--
Stop
2620

FIG. 27

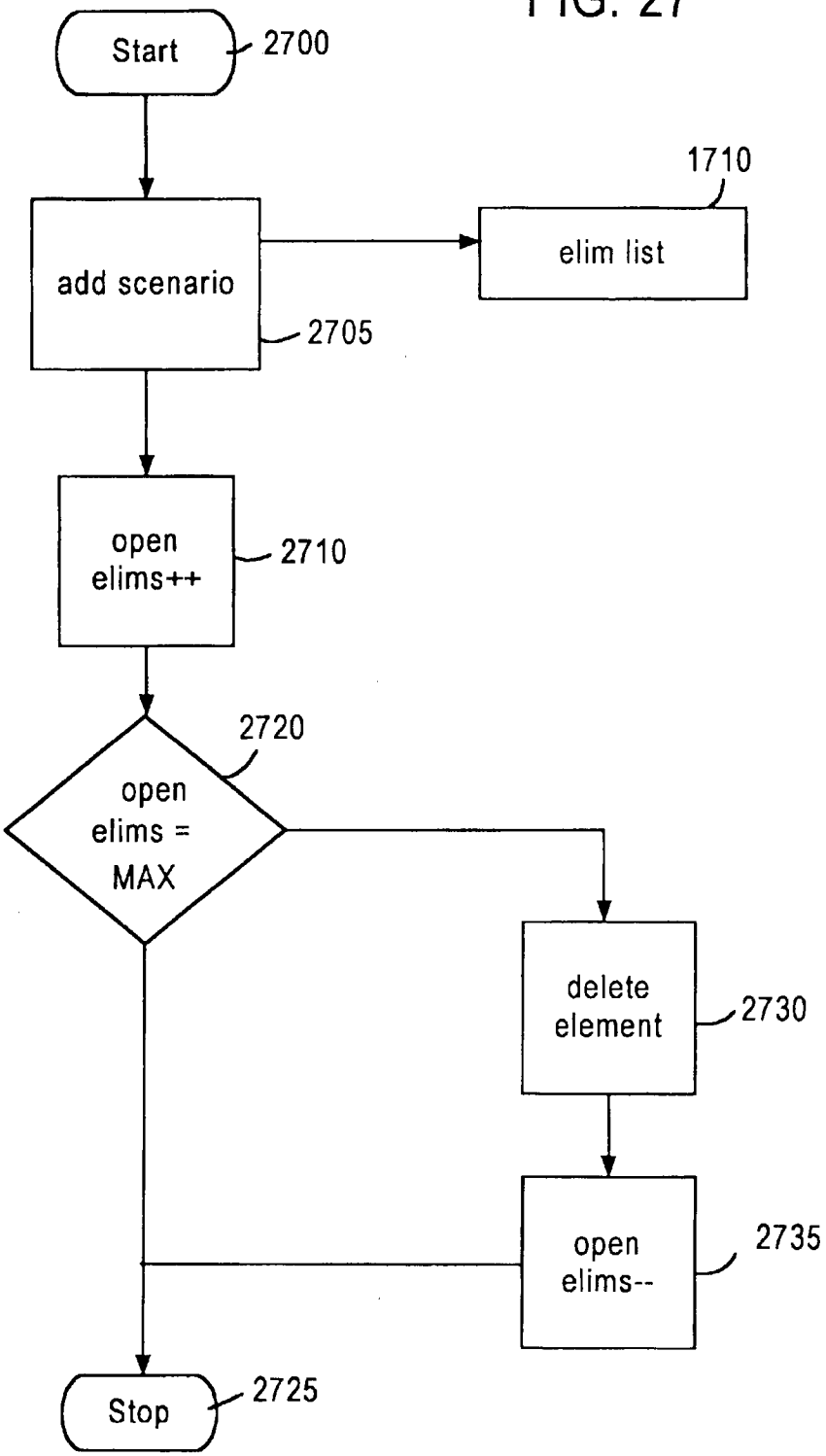

Start
2700
1710
add scenario
elim list
2705
open
elims++
2710
2720
open
elims =
MAX
delete
element
2730
open
elims--
2735
Stop
2725

FIG. 28

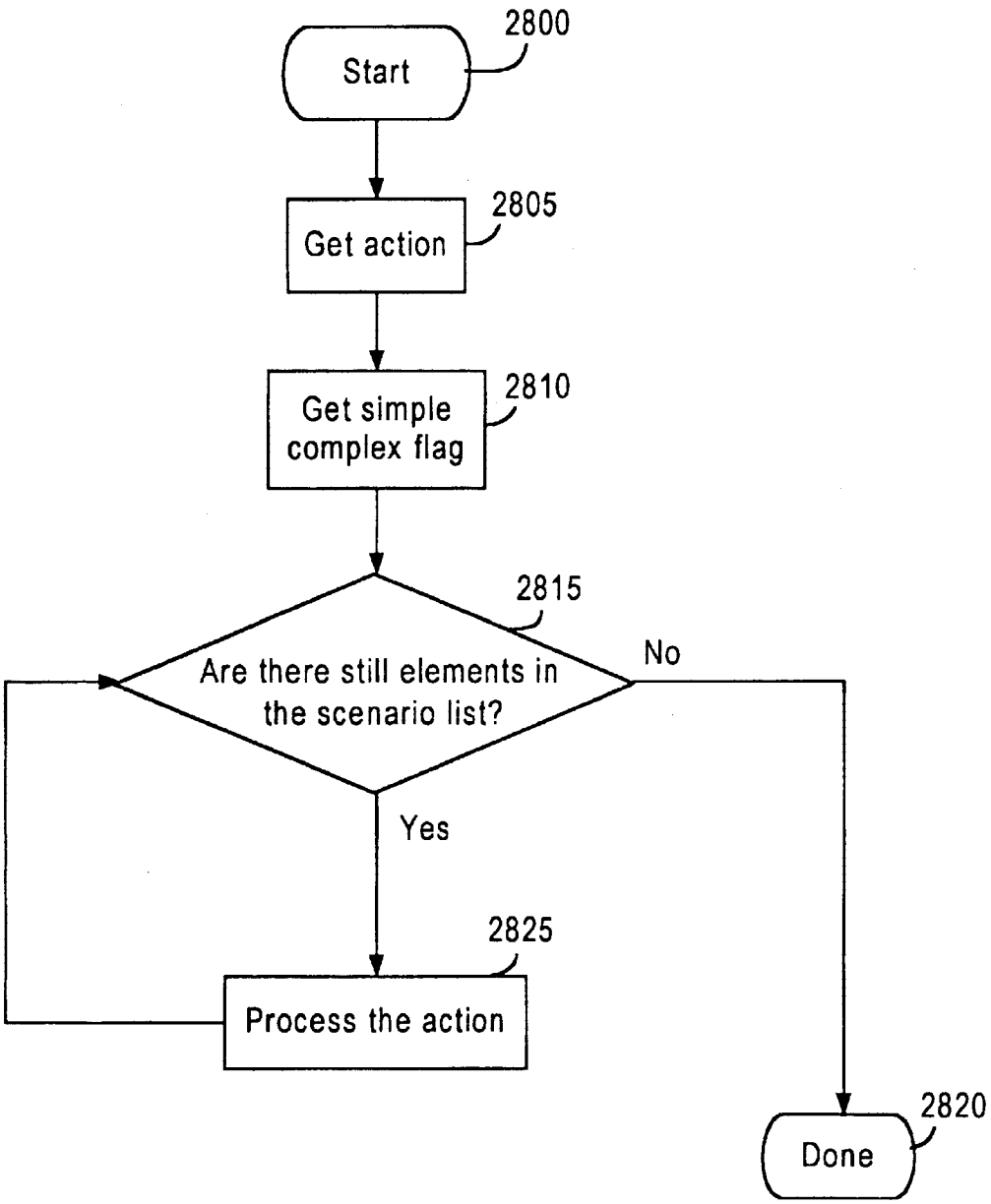

2800
Start
2805
Get action
2810
Get simple complex flag
2815
Are there still elements in the scenario list?
No
Yes
2825
Process the action
2820
Done

FIG. 29

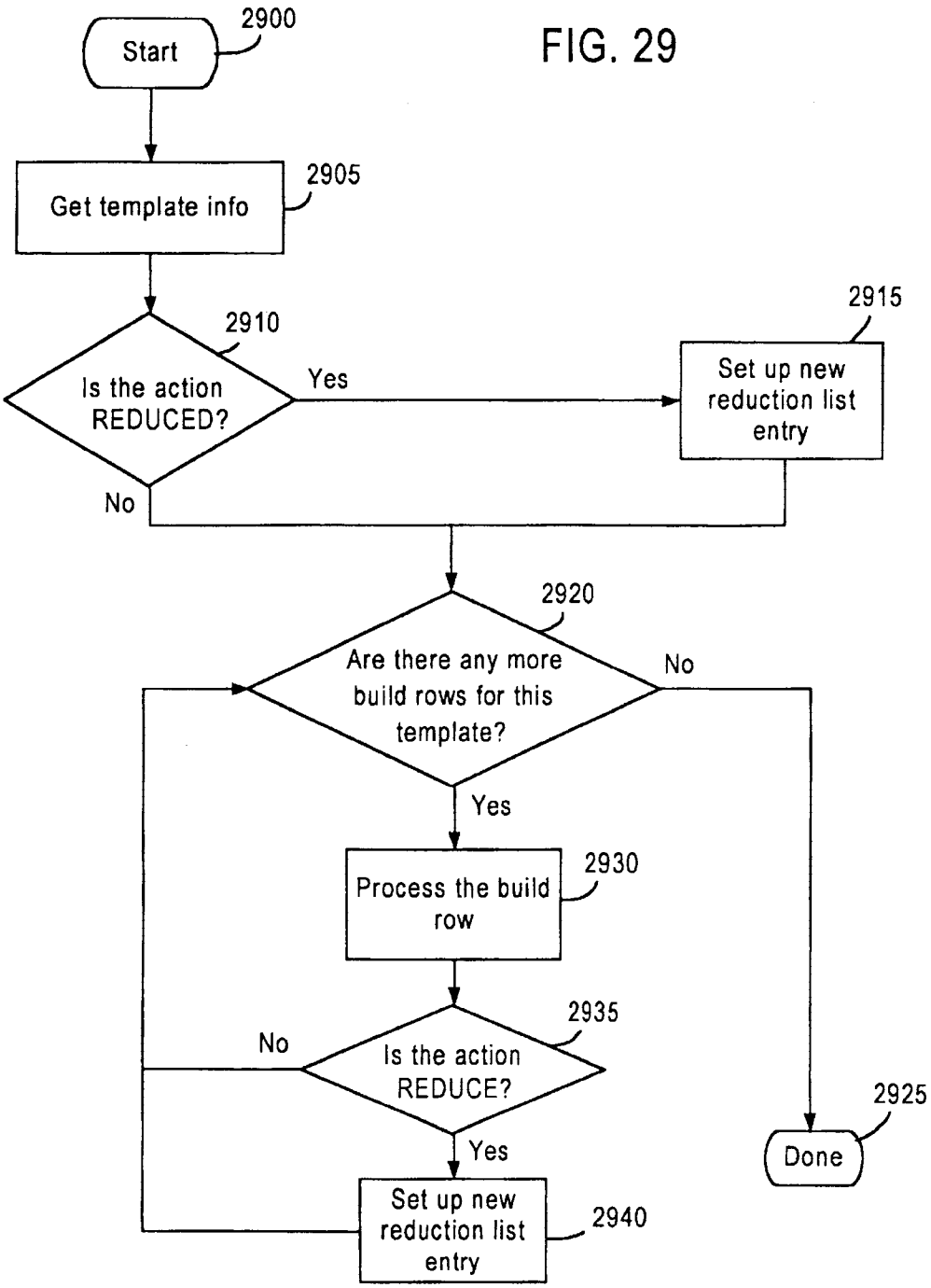

Start
2900
Get template info
2905
Is the action REDUCED?
2910
Yes
No
Set up new reduction list entry
2915
Are there any more build rows for this template?
2920
No
Yes
Process the build row
2930
Is the action REDUCE?
2935
No
Yes
Set up new reduction list entry
2940
Done
2925

FIG. 30

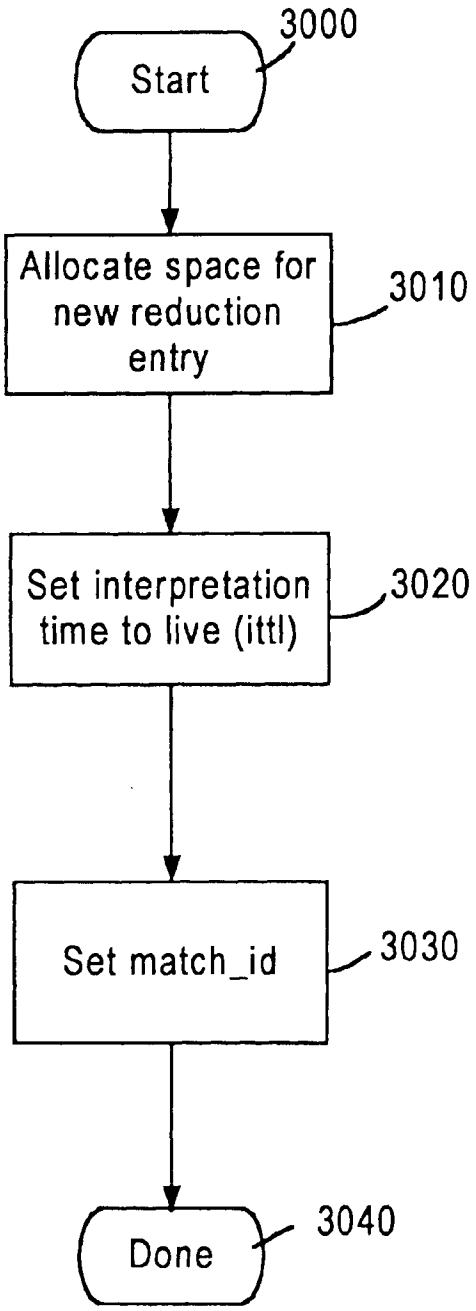

3000
Start
Allocate space for new reduction entry
3010
Set interpretation time to live (ittl)
3020
Set match_id
3030
Done
3040

FIG. 31

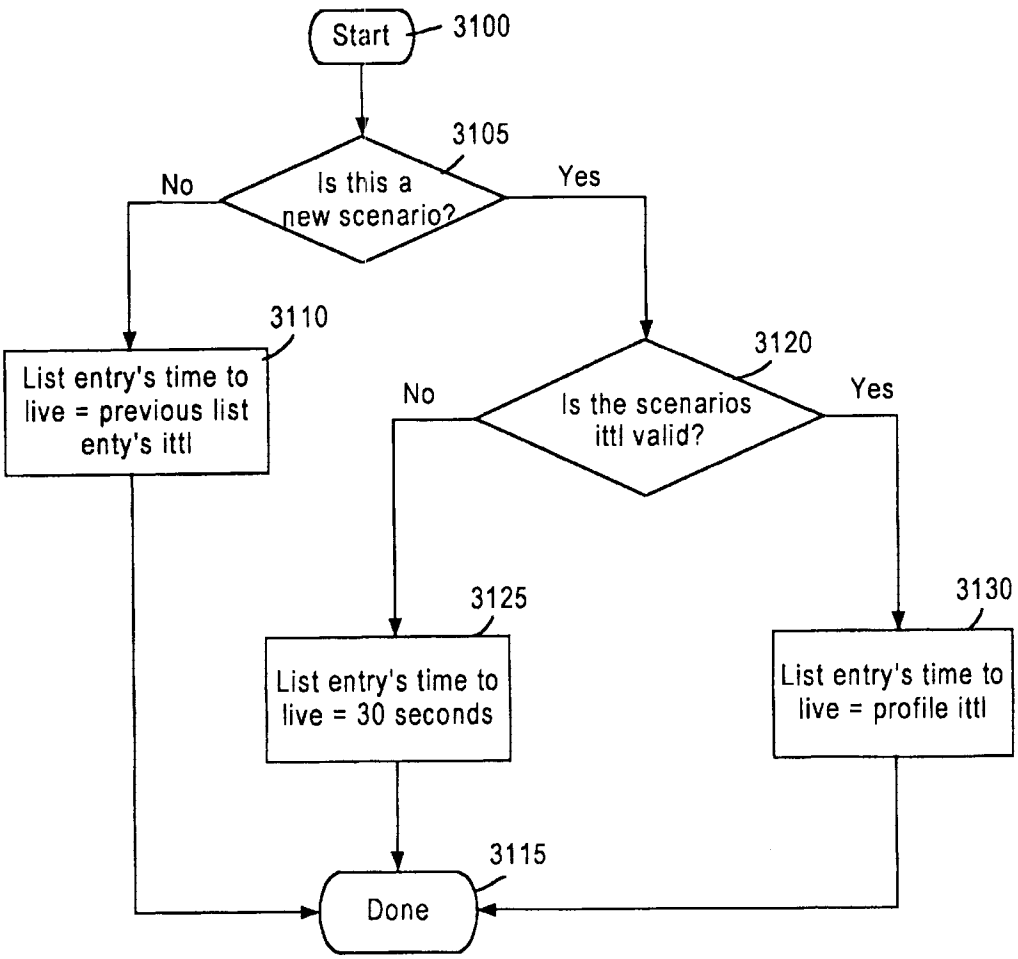

Start
3100
3105
Is this a new scenario?
No
Yes
3110
List entry's time to live = previous list enty's ittl
3120
Is the scenarios ittl valid?
No
Yes
3125
List entry's time to live = 30 seconds
3130
List entry's time to live = profile ittl
3115
Done

FIG. 32

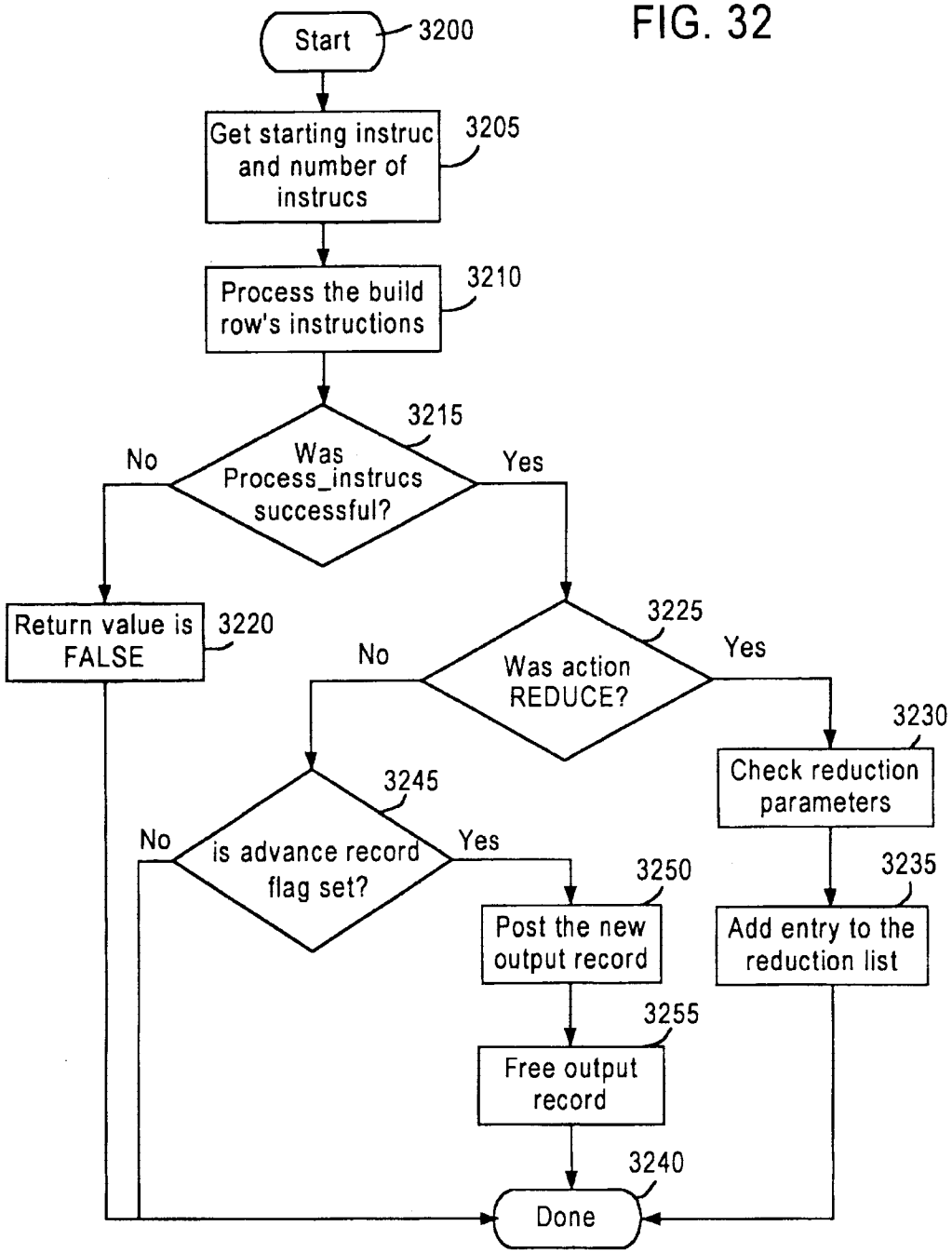

Start
3200
Get starting instruc and number of instrucs
3205
Process the build row's instructions
3210
3215
Was Process_instrucs successful?
No
Yes
Return value is FALSE
3220
3225
Was action REDUCE?
No
Yes
3230
Check reduction parameters
3245
is advance record flag set?
No
Yes
3250
Post the new output record
3235
Add entry to the reduction list
3255
Free output record
3240
Done

FIG. 33

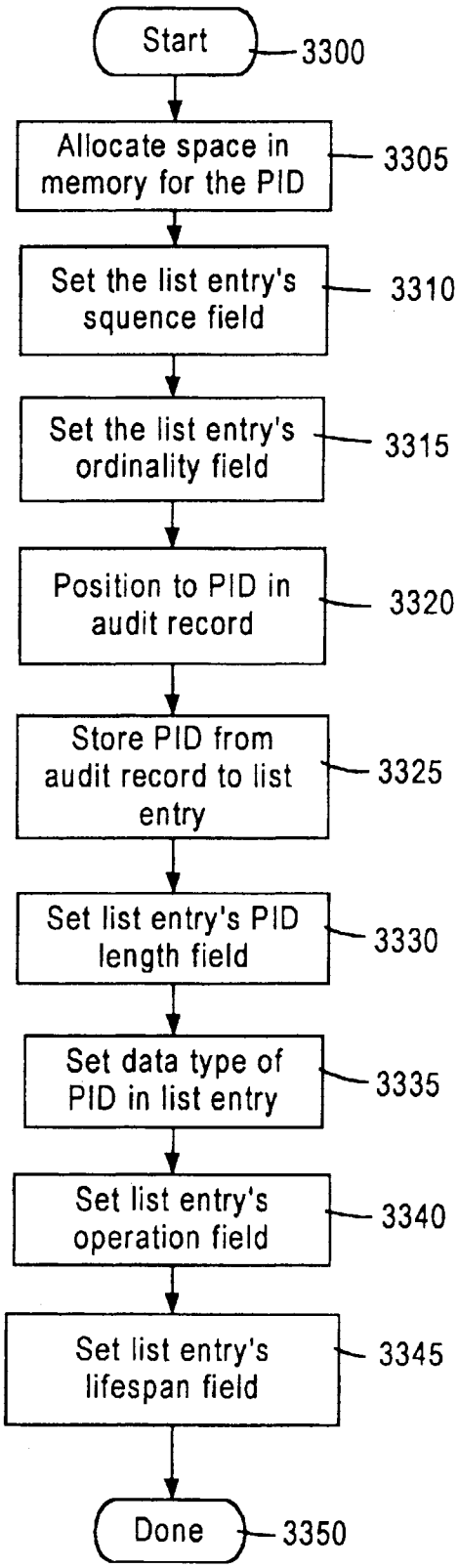

Start
3300
Allocate space in memory for the PID
3305
Set the list entry's squence field
3310
Set the list entry's ordinality field
3315
Position to PID in audit record
3320
Store PID from audit record to list entry
3325
Set list entry's PID length field
3330
Set data type of PID in list entry
3335
Set list entry's operation field
3340
Set list entry's lifespan field
3345
Done
3350

FIG.34

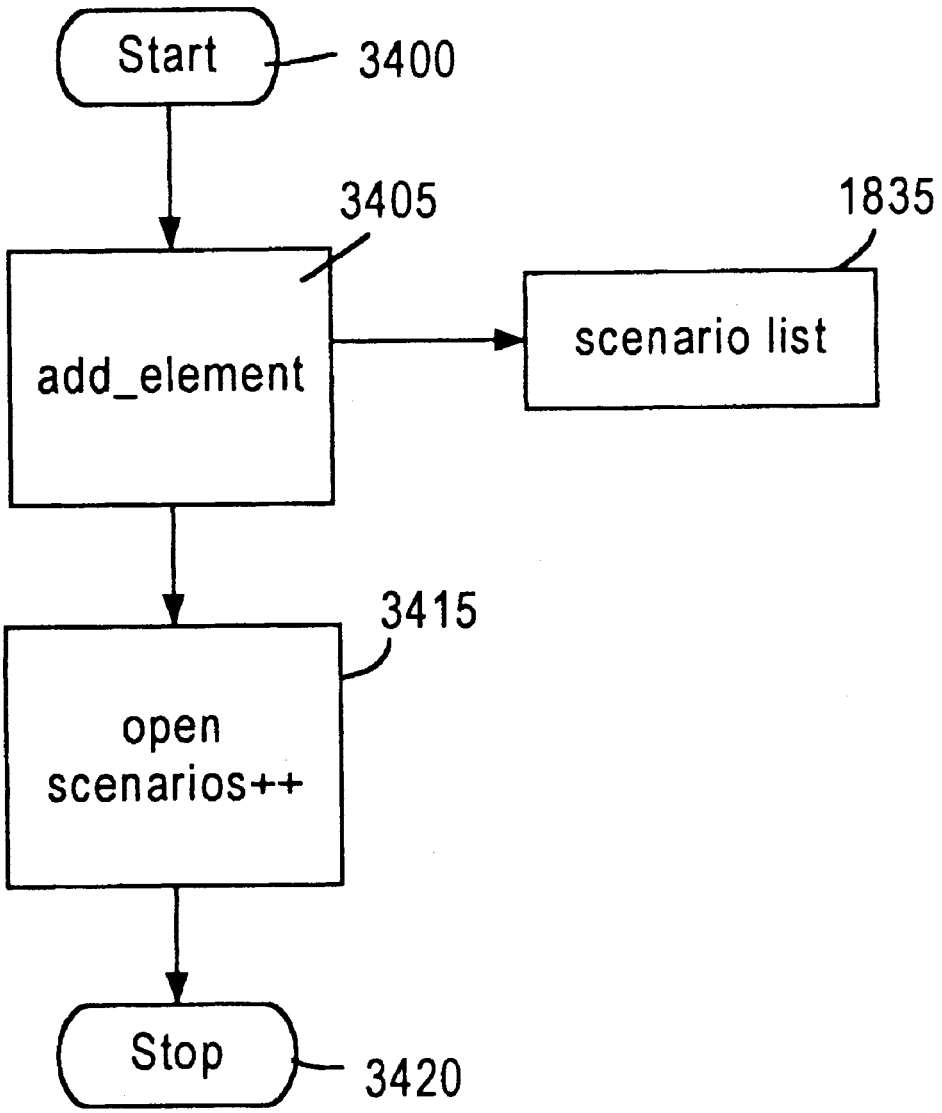
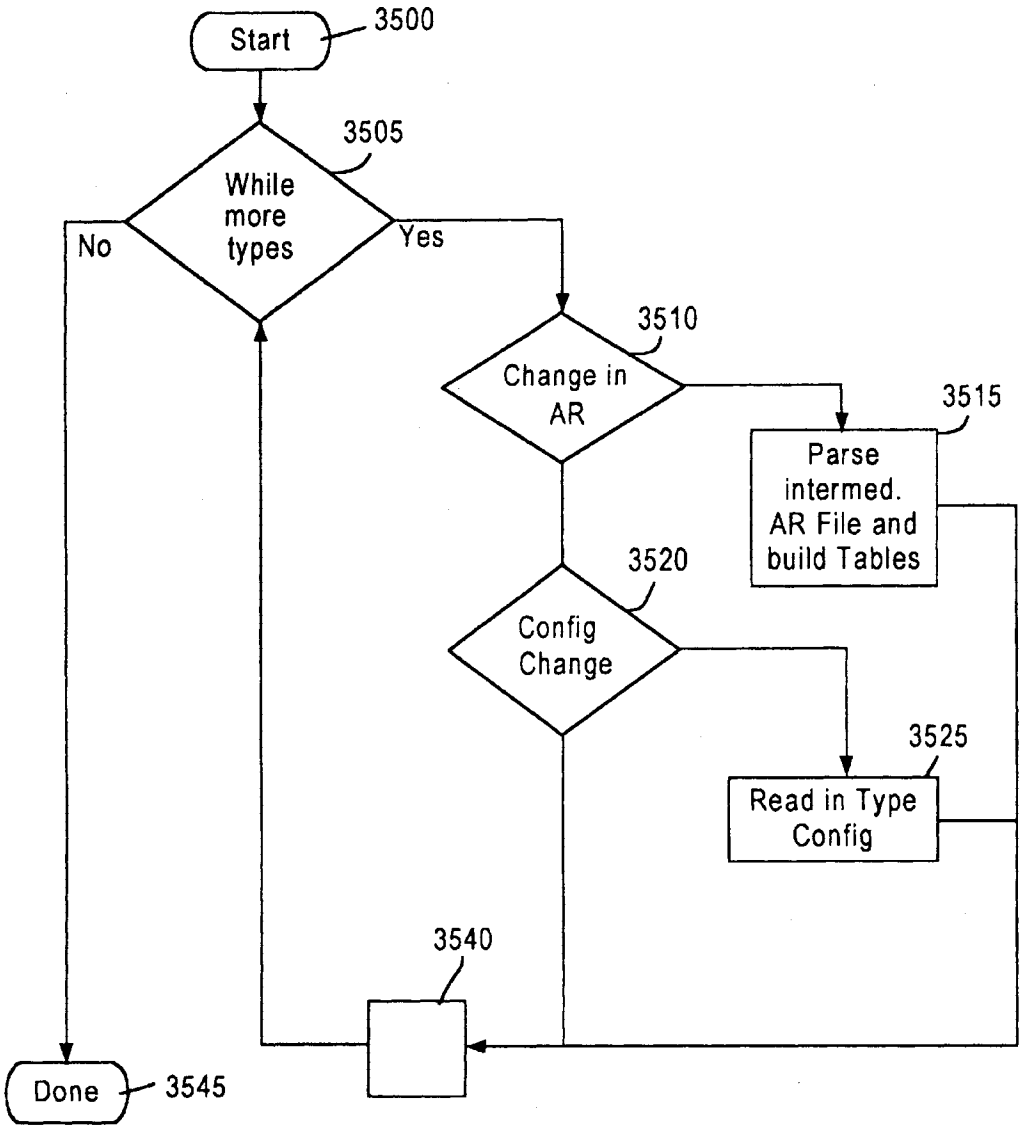

Start
3400
3405
1835
add_element
scenario list
3415
open
scenarios++
Stop
3420

FIG. 35


Start
3500
3505
While more types
No
Yes
3510
Change in AR
3515
Parse intermed. AR File and build Tables
3520
Config Change
3525
Read in Type Config
3540
Done
3545

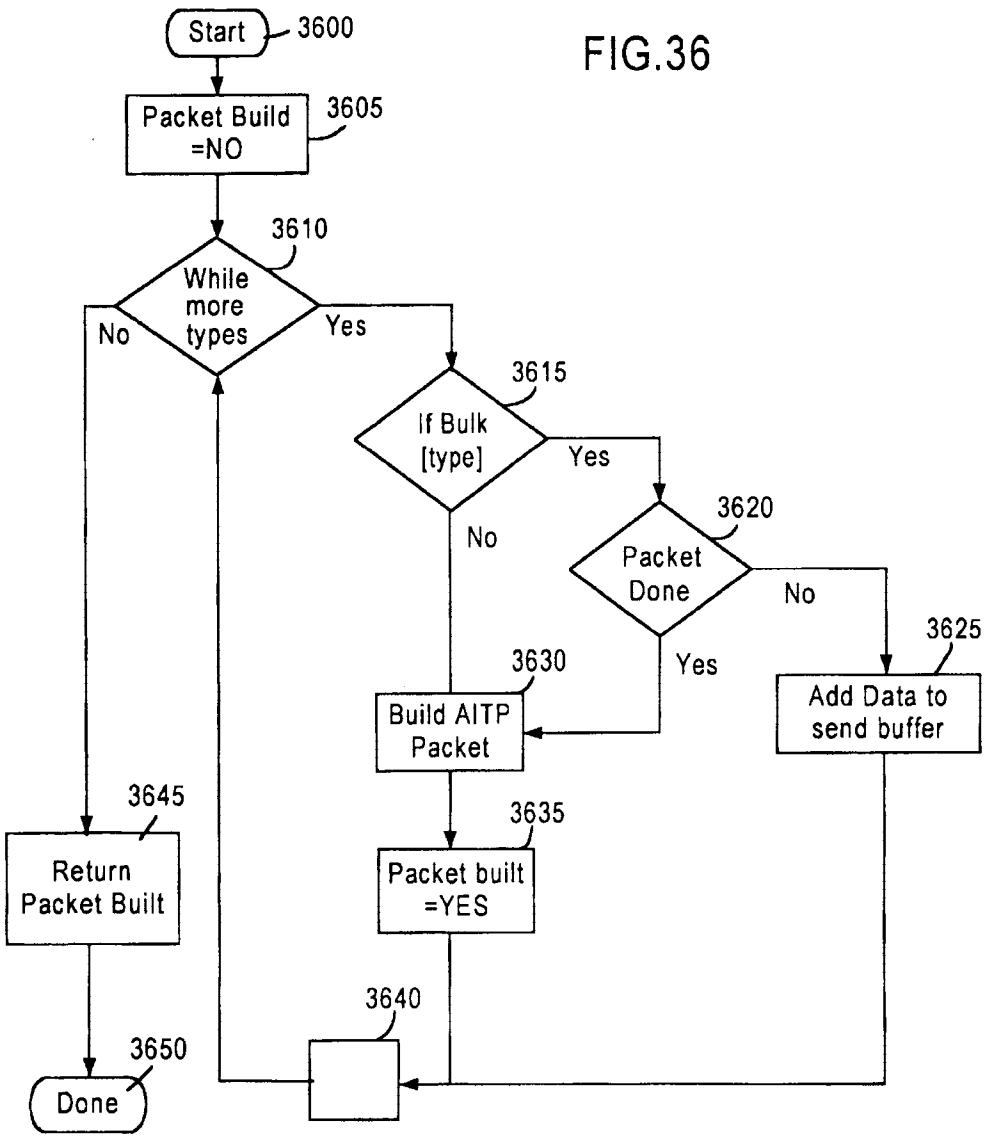

Start
3600
Packet Build =NO
3605
3610
While more types
No
Yes
3615
If Bulk [type]
Yes
No
3620
Packet Done
No
Yes
3625
Add Data to send buffer
3630
Build AITP Packet
3635
Packet built =YES
3640
3645
Return Packet Built
3650
Done

FIG.36

FIG.37

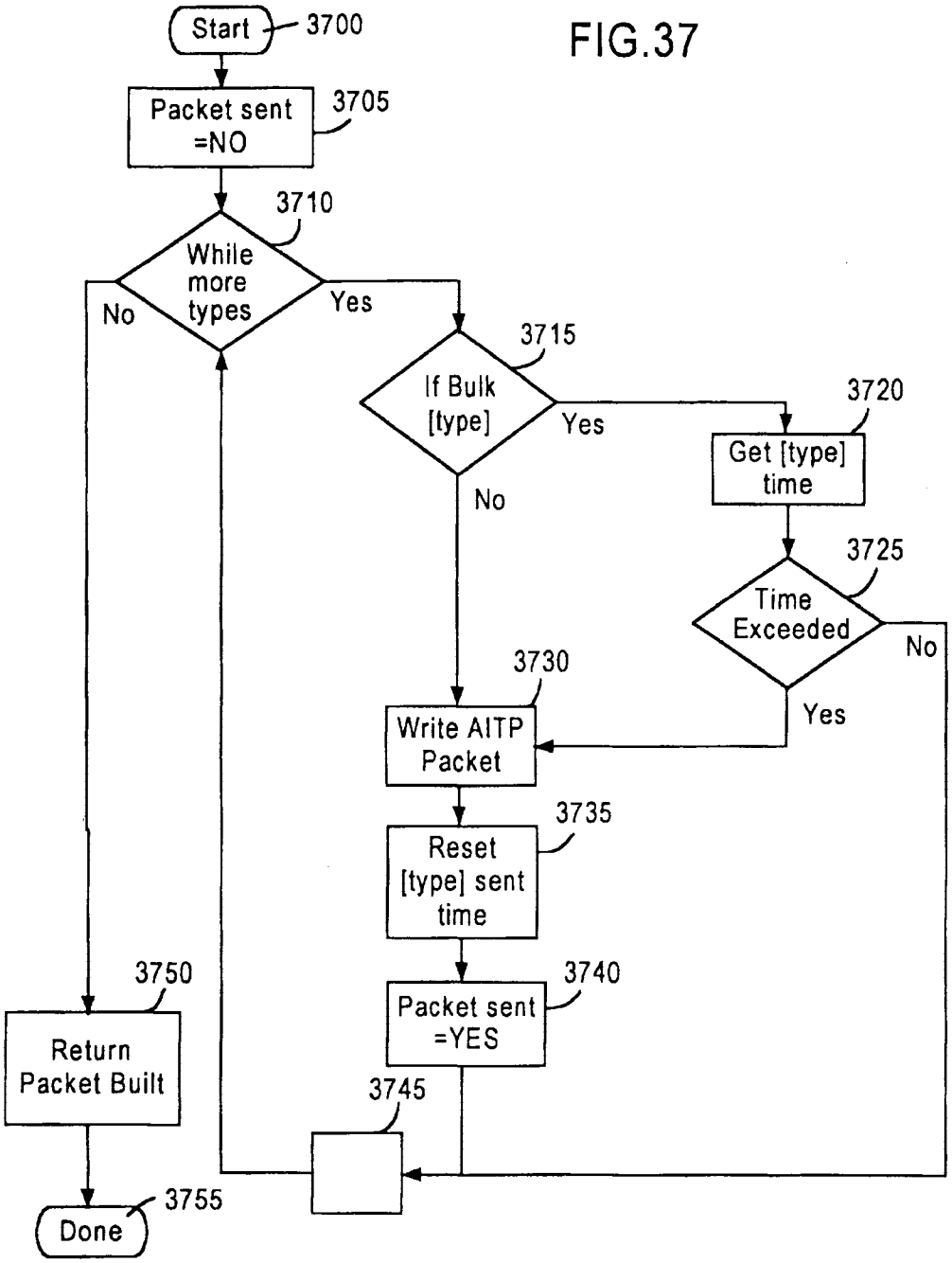

Start
3700
Packet sent =NO
3705
3710
While more types
No
Yes
3715
If Bulk [type]
Yes
No
3720
Get [type] time
3725
Time Exceeded
No
Yes
3730
Write AITP Packet
3735
Reset [type] sent time
3740
Packet sent =YES
3745
3750
Return Packet Built
Done
3755

FIG. 38

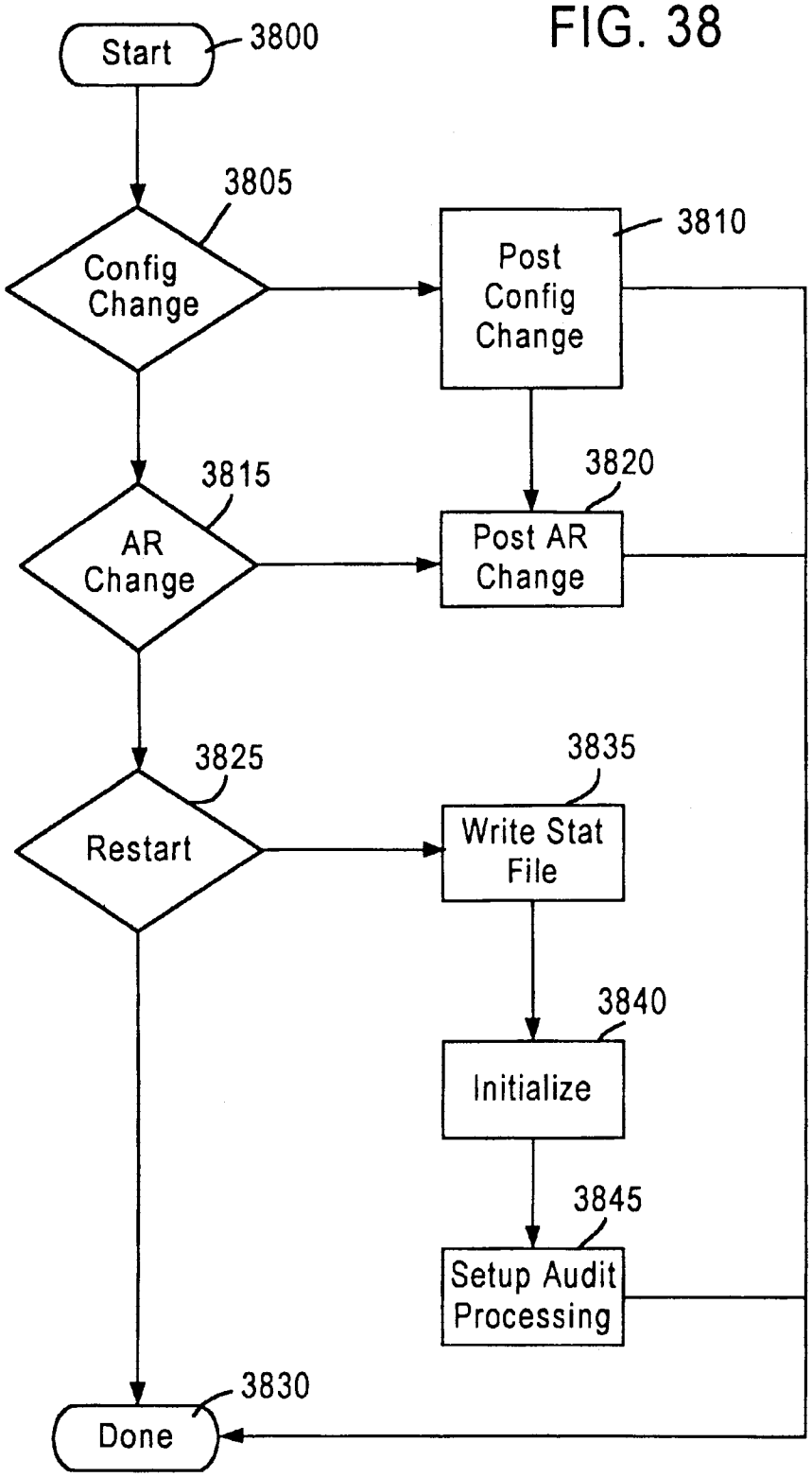
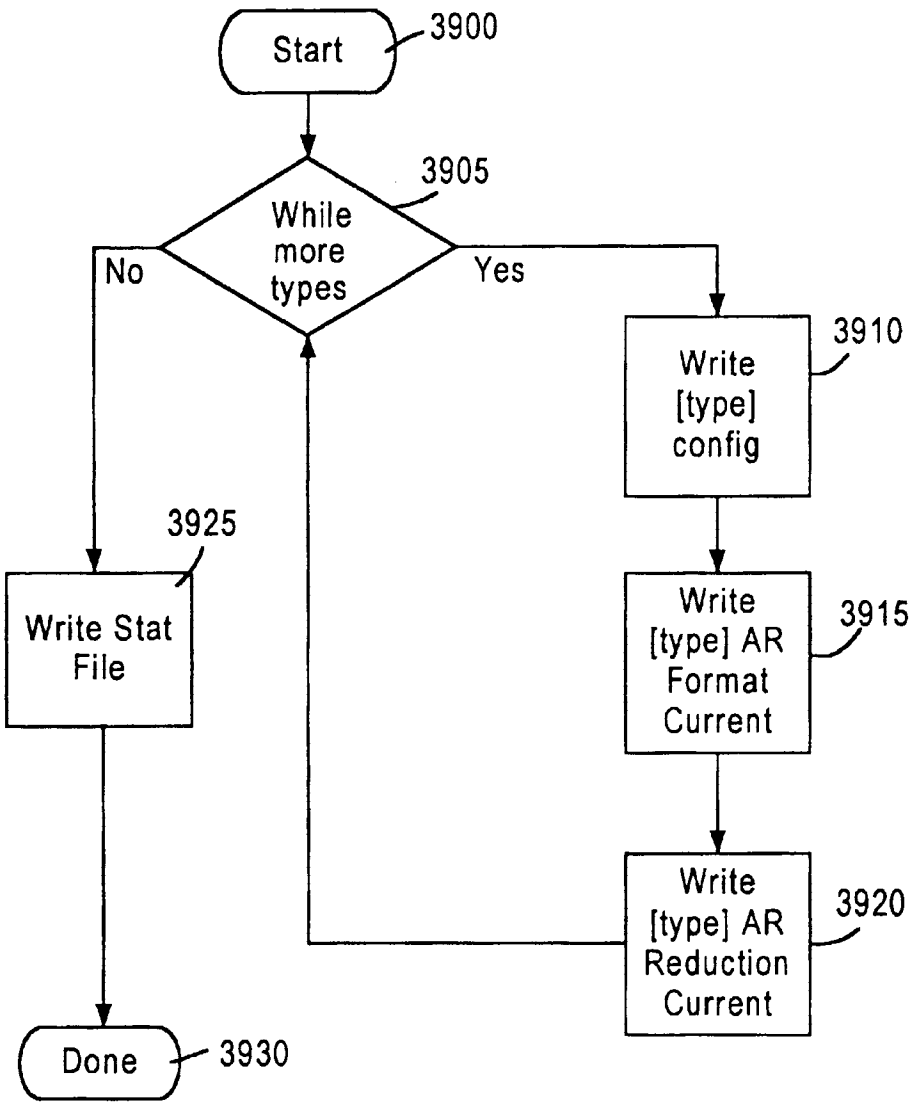

Start
3800
3805
Config Change
3810
Post Config Change
3815
AR Change
3820
Post AR Change
3825
Restart
3835
Write Stat File
3840
Initialize
3845
Setup Audit Processing
3830
Done

FIG. 39


Start
3900
While more types
3905
No
Yes
Write [type] config
3910
Write [type] AR Format Current
3915
Write [type] AR Reduction Current
3920
Write Stat File
3925
Done
3930

METHOD AND SYSTEM FOR REDUCING THE VOLUME OF AUDIT DATA AND NORMALIZING THE AUDIT DATA RECEIVED FROM HETEROGENEOUS SOURCES

FIELD OF THE INVENTION

The present invention relates generally to intrusion detection systems for computer systems, and more particularly, to a method and apparatus for analyzing audit data received from heterogeneous sources, reducing the volume of the analyzed audit data from further consideration, normalizing audit trail records received from the heterogeneous sources to a standardized format, and performing an initial review of the normalized audit trail records. Based upon the initial review, intrusions can be detected and alerts provided.

BACKGROUND OF THE INVENTION

The development of the computer and its astonishingly rapid improvement have ushered in the Information Age that affects almost all aspects of commerce and society. Just like the physical infrastructures that support the American economy, there is a highly developed computer infrastructure that supports the American and worldwide economy.

Besides traditional physical threats to United States security, the security of the United States is also dependent on protecting the computer infrastructure that supports American government and industry. The computer infrastructure is open to attack by hackers and others, who could potentially wreak havoc.

The President of the United States has recognized the existence of these infrastructures and has created the President's Commission on Critical Infrastructure Protection. This Commission was constituted to determine which industries are critical and whether these industries were vulnerable to cyber attack. The Commission issued a report and deemed transportation, oil and gas production and storage, water supply, emergency services, government services, banking and finance, electrical power and telecommunications to be critical infrastructures which rely on the computer infrastructure.

A personal computer and a modem with access to the Internet are all the tools that a computer hacker needs to conduct a cyber attack on a computer system. The rapid growth of a computer-literate population ensures that millions of people possess the skills necessary to consider a cyber attack. The computer literate population includes recreational hackers who attempt to gain unauthorized electronic access to information and communication systems. These computer hackers are often motivated only by personal fascination with hacking as an interesting game. Criminals, and perhaps organized crime, might also attempt personal financial gain through manipulation of financial or credit accounts or stealing services. Industrial espionage can also be the reason for a cyber attack on a competitor's computer system. Terrorists may attempt to use the computer infrastructure. Other countries may use the computer infrastructure for national intelligence purpose. Finally, there is the prospect of information warfare, which is a broad, orchestrated attempt to disrupt a United States military operation or significant economic activity.

A typical secure computer network has an interface for receiving and transmitting data between the secure network and computers outside the secure network. A plurality of network devices are typically behind the firewall. The interface may be a modem or an Internet Protocol (IP) router.

Data received by the modem is sent to a firewall which is a network security device that only allows data packets from a trusted computer to be routed to specific addresses within the secure computer network. Although the typical firewall is adequate to prevent outsiders from accessing a secure network, hackers and others can often breach a firewall. This can occur by cyber attack where the firewall becomes overwhelmed with requests and errors are made permitting access to an unauthorized user. As can be appreciated, new ways of overcoming the security devices are developed everyday. An entry by an unauthorized computer into the secured network, past the firewall, from outside the secure network is called an intrusion. This is one type of unauthorized operation on the secure computer network.

Another type of unauthorized operation is called a misuse. A misuse is an unauthorized access by a computer within the secure network. In a misuse situation, there is no breach of the firewall. Instead, a misuse occurs from inside the secure computer network. A misuse can be detected when an authorized user performs an unauthorized, or perhaps, infrequent operation which may raise the suspicion that the authorized user's computer is being misused. For example, an unauthorized user could obtain the password of an authorized user and logon to the secured network from the authorized computer user's computer and perform operations not typically performed by the authorized user. Another example might be where a terrorist puts a gun to the head of an authorized user and directs the authorized user to perform unauthorized or unusual operations.

There are systems available for determining that a breach of computer security has occurred. These systems can broadly be termed intrusion detection systems. Existing intrusion detection systems can detect intrusions and misuses. The existing security systems determine when computer misuse or intrusion occurs. Computer misuse detection is the process of detecting and reporting uses of processing systems and networks that would be deemed inappropriate or unauthorized if known to responsible parties. An intrusion is an entry to a processing system or network by an unauthorized outsider.

Processing system misuse detection and reporting research has been funded by U.S. government agencies that have concerns for the confidentiality of their computer systems. Researchers have generally been associated with large research organizations or national laboratories. These institutions have required detailed knowledge of technical computer security, known threats and vulnerabilities, protection mechanisms, standard operational procedures, communications protocols, details of various systems' audit trails, and legal investigation of computer crimes. This misuse detection and reporting research has followed two basic approaches: anomaly detection systems and expert systems.

Anomaly detection systems look for statistically anomalous behavior.

These systems assume that intrusions and other security problems are rare and that they appear unusual when compared to other user behavior. D. Denning, "An Intrusion Detection Model," Proc 1986 IEEE Symp. Security & Privacy (April 1986) provides an anomaly detection model (hereinafter the "Denning Model") for detecting intrusions into computer systems. The Denning Model uses statistical scenarios for user, dataset, and program usage to detect "exceptional" use of the system.

There are variations of the Denning Model and different applications of these various models. Anomaly detection

techniques such as those based on the Denning Model, however, have generally proven to be ineffective and inefficient. Anomaly detection techniques, for instance, do not detect most actual misuses. The assumption that computer misuses would appear statistically anomalous has been proven false. When scripts of known attacks and misuses are replayed on computers with statistical anomaly detection systems, few if any of the scripts are identified as anomalous. This occurs because the small number of commands in these scripts are insufficient to violate profiling models.

In general, anomaly detection techniques cannot detect particular instances of misuses unless the specific behaviors associated with those misuses also satisfy statistical tests without security relevance. Anomaly detection techniques also produce false alarms. Most of the reported anomalies are purely statistical and do not reflect security problems. These false alarms often cause system managers to resist using anomaly detection method because they increase the processing system workload without substantial benefits.

Another limitation with anomaly detection approaches is that users activities are often too varied for a single scenario and can result in many false alarms. Statistical measures also are not sensitive to the order in which events occur, and this may prevent detection of serious security violations that exist when events occur in a particular order. Scenarios that anomaly detection techniques use also may be vulnerable to conscious manipulation by users. Consequently a knowledgeable perpetrator may train the thresholds of detection system adaptive scenarios to accept aberrant behaviors as normal. Furthermore, statistical techniques that anomaly detection systems use require complicated mathematical calculations and, therefore, are usually computationally expensive.

Expert systems (also known as rule-based systems or production system) have had some use in misuse detection, generally as a layer on top of anomaly detection systems for interpreting reports of anomalous behavior. Since the underlying model was anomaly detection, they have the same drawbacks of anomaly detection techniques.

Expert system approaches, in addition, are themselves inherently inefficient. S. Snapp, et al., "DIDS (Distributed Intrusion Detection System)" Proc. 14th Nat'l Computer Security Conf., Washington, D.C. (October 1991) describes one example of an expert system signature analysis model that detects misuse by looking for one specific event within a specific system context. In one study, this detection system was found to be two and four orders of magnitude slower than "hard-wired" techniques and much too slow for real-time operation. This also makes it impractical to use these systems to detect and report misuses of multiple associated processing systems through operation of a single misuse detection and reporting system.

Expert systems approaches are also not deterministic. Consequently, these rules are expressed in a declarative, non-procedural fashion. When rule changes occur, it is generally extremely difficult to predict how the new system will behave. This makes development and testing more complex and expensive. Moreover, expert system approaches are limited to the knowledge of the expert who programmed the rules into the system. However, an expert is only capable of programming the rules of behavior that the expert knows. Since there are often many different paths to a particular misuse, the expert will be unable to create rules that represent all of these paths.

More recent attempts at detecting misuse have relied on a signature mechanism with a signature being the set of events and transitions functions that define the sequence of actions that form a misuse. A misuse engine that uses this signature mechanism is described in detail in U.S. Pat. No. 5,557,742. The signature mechanism uses audit trail records typically generated by computer operating systems. The user selects a plurality of misuses that together form the signature mechanism. Although the signature mechanism goes a step beyond expert systems, it is similar to an expert system because it relies upon signatures or rules.

A need exists for an intrusion detection system which can provide early warning of potential misuses and intrusions without relying on particular rules or signatures which can be easily thwarted. Early warning can be provided by eliminating most of the audit trail records before a misuse and intrusion detection engine further analyzes the audit trail records.

SUMMARY OF THE INVENTION

It is, therefore, an object of the present invention to provide a method and apparatus for eliminating audit trail records from further consideration by an intrusion and misuse detection engine.

It is a further object of the present invention to normalize a portion of the audit trail records received from heterogeneous sources for further analysis by the misuse and intrusion detection engine.

A further object is to provide an early warning of a potential misuse or intrusion.

Another object is to use a weighting system using barriers and boundaries so that a potential misuse or intrusion is identified before it occurs.

An additional object is to provide an intrusion detection system in which particular misuses do not have to be known in advance and which does not require rule sets or signatures of particular misuses.

These and other objects of the present invention are achieved by a method of reducing the volume of native audit data from further analysis by a misuse and intrusion detection engine. Typically, more than ninety percent of the volume of audit information received from heterogeneous operating systems does not need to be analyzed by a misuse and intrusion detection engine because this audit information can be filtered out as not posing a security threat. Advantageously, by reducing (eliminating) the volume of audit information, a misuse and intrusion engine can more quickly determine whether a security threat exists because the volume of data that the engine must consider is drastically reduced. Also, advantageously, the audit information that is forwarded to the engine is normalized to a standard format, thereby reducing the computational requirements of the engine. The method of reducing the volume of native audit data includes comparing each of the native audits against at least one template and against at least one native audit. By matching the native audits against templates of native audits that do not pose security threats, the native audits that do not pose security threats can be reduced out from further consideration. The native audits that are determined to pose potential security threats are transformed into a standardized format for further analysis by a misuse and intrusion detection engine.

The foregoing objects are also achieved by a method of reducing the volume of native audits received from at least one operating system with each of the native audits being in a particular format. The particular format is identified for each of the received native audits. Each of the received identified native audits are compared against at least one

template and it is determined if each of the native audits matches at least one template. Each of the matched native audits is reduced.

The foregoing objects are also achieved by an article including at least one sequence of machine executable instruction on a medium bearing the executable instructions in machine readable form. Execution of the instructions by one or more processors causes the one or more processors to identify the particular format for each of the received native audits. The one or more processors compares each of the received identified native audits against at least one template and determines if each of the native audits matches at least one template. The one or more processors reduces each of the matched native audits.

The foregoing objects are also achieved by a computer architecture for reducing the volume of native audits received from at least one operating system with each of the native audits being of particular format. The computer architecture includes identifying means for identifying a particular format for each of the received native audits. Comparing means are provided for comparing each of the received identified native audits against at least one template and determining if each of the native audits matches at least one template. Reducing means are provided for reducing each of the native audits.

The foregoing objects are also achieved by a computer system including a processor and a memory coupled to the processor, the memory having stored therein sequences of instructions, which, when executed by the processor, causes the processor to perform the steps of identifying a particular format for each of the received identified native audits. The processor performs the step of comparing each of the received identified native audits against at least one template. The processor determines if each of the native audits matches at least one template and reduces each of the matched audits.

Still other objects and advantage of the present invention will become readily apparent to those skilled in the art from following detailed description, wherein the preferred embodiments of the invention are shown and described, simply by way of illustration of the best mode contemplated of carrying out the invention. As will be realized, the invention is capable of other and different embodiments, and it several details are capable of modifications in various obvious respects, all without departing from the invention. Accordingly, the drawings and description thereof are to be regarded as illustrative in nature, and not as restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by limitation, in the figures of the accompanying drawings, wherein elements having the same reference numeral designations represent like elements throughout and wherein:

FIG. **1** is a high-level block diagram of an exemplary secured computer network on which the present invention can be used;

FIG. **2** is a high-level block diagram of an exemplary computer system with which the present invention can be used;

FIG. **3** is a high-level block diagram of the hardware used for an audit reduction agent;

FIG. **4** is an illustration of a logical architecture used for the audit reduction agent;

FIG. **5**A is an illustration of a native audit;

FIG. **5**B is an illustration of audits is a diagram of an embodiment of the invention as used in several fleets of trucks in a wireless network;

FIG. **6** is an illustration of a matrix of templates;

FIGS. **7**A–**7**C are illustrations of simple, simple-complex and complex—complex comparisons and interpretations, respectively;

FIG. **7**D is a flow diagram of the reduction process of the present invention; and

FIGS. **8**–**39** are flow diagrams of the process used for analyzing and reducing audit trail records from heterogeneous sources according to the present invention.

BEST MODE FOR CARRYING OUT THE INVENTION

A method and apparatus for normalizing audit trail records received from heterogeneous sources to a standardized format, performing an initial review of the normalized audit trail records and reducing the number of audit trail records to be stored for later review are described. Based upon the initial review, intrusions can be detected and alerts provided. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention. The present invention is usable in conjunction with a patent application entitled "Dynamic System Defense for Information Warfare" assigned to the instant assignee and filed on even date herewith and incorporated by reference into this specification in its entirety.

To a great extent, monitoring of user and process behavior (and detection of anomalies in that behavior) can be based on security-relevant audit information. Audited systems produce a huge amount of audit data which can be analyzed and interpreted to determine whether security breaches are occurring or have occurred. Analysis allows an Information Security Officer (ISO) to recognize suspicious behavior and to respond to the behavior quickly enough to effectively counter a possible security threat. The move towards distributed data handling systems requires security monitoring to ascertain a system wide security posture as opposed to determining that there is a potential security threat for a single computer. This means that a hacker may attack two distributed computers simultaneously and the ISO needs to be aware that a security threat exists. The ISO can then identify when people inadvertently or purposely exploit the systems security weaknesses. The present invention addresses the following specific needs:

1) reduces the volume of security relevant audit data,
2) performs timely audit analysis,
3) automates detection of suspicious behavior,
4) works with minimal user involvement,
5) helps define a model for security monitoring, and
6) allows an organization to respond to an evolving variety of threats.

FIG. **1** is a block diagram illustrating an exemplary computer network **100** including a plurality of network devices on which an embodiment of the invention may be implemented. According to the present invention, an audit agent can reside on one or all of the nodes depicted on the computer network **100**. As explained below, the audit agent resides on audit server **120** and receives audit trail informa-

tion from each of the nodes on the computer network **120**. The network devices include devices such as hosts, servers and personal computers. The present invention is usable on such networks as ARCnet, Ethernets and Token-Ring networks, wireless networks, among other networks. The network **100**, in this example, has a central network cable **102**, also known as media, which may be of any known physical configuration including unshielded twisted pair (UTP) wire, coaxial cable, shielded twisted pair wire, fiber optic cable, and the like. Alternatively, the network devices could communicate across wireless links.

The network **100** includes a network server **104** coupled to the network cable **102** and another server **106** coupled to the network cable **102**. A host computer **108** is coupled to the network cable **102**. A terminal **110** is coupled to the network cable **102**. A personal computer **112** is coupled to the network cable **102**. Each network device **104, 106, 108, 110, 112** can also be considered a node because each device has an addressable interface on the network. As can be appreciated, many other devices can be coupled to the network including additional personal computers, mini-mainframes, mainframes and other devices not illustrated or described which are well known in the art.

A security server **114** is coupled to the network cable **102**. A firewall **116** connects the secure network **100** to an interface **118**. The firewall **116** is a combination hardware and software buffer that is between the internal network **100** and external devices outside the internal computer network **100**. The network devices within the internal network **100** appear within the dashed lines in FIG. **1**, and the external devices outside the internal network appear outside the dashed lines in FIG. **1**. The firewall **116** allows only specific kinds of messages from external devices to flow in and out of the internal network **100**. As is known, firewalls are used to protect the internal network **100** from intruders or hackers who might try to break into the internal network **100**. The firewall **116** is coupled to an interface **118**. The interface **118** is external to the network **100** and can be a modem or an Internet Protocol (IP) router and serves to connect the secure network **100** to devices outside the secure network. An audit server is depicted at **120**. For illustrative purposes, an intruder computer system is depicted at **130**.

FIG. **2** is a block diagram illustrating an exemplary computer system, such as the audit server **120** depicted in FIG. **1**, usable on the internal secure network **100**. The present invention is usable with currently available personal computers, mini-mainframes, mainframes and the like. Although audit server **120** is depicted in FIG. **1** as a network device which is part of a wired local network, the audit server **120** is also envisioned as being connected to the network **100** by a wireless link.

Audit server **120** includes a bus **202** or other communication mechanism for communicating information, and a processor **204** coupled with the bus **202** for processing information. Audit server **120** also includes a main memory **206**, such as a random access memory (RAM) or other dynamic storage device, coupled to the bus **202** for storing information and instructions to be executed by processor **204**. Main memory **206** also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor **204**. Audit server **120** further includes a read only memory (ROM) **208** or other static storage device coupled to the bus **202** for storing static information and instructions for the processor **204**. A storage device **210**, such as a magnetic disk or optical disk, is provided and coupled to the bus **202** for storing information and instructions.

Audit server **120** may be coupled via the bus **202** to a display **212**, such as a cathode ray tube (CRT) or a flat panel display, for displaying information to a computer user. An input device **214**, including alphanumeric and other keys, is coupled to the bus **202** for communicating information and command selections to the processor **204**. Another type of user input device is cursor control **216**, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor **204** and for controlling cursor movement on the display **212**. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y) allowing the device to specify positions in a plane.

The processor **204** can execute sequences of instructions contained in the main memory **206**. Such instructions may be read into main memory **206** from another computer-readable medium, such as storage device **210**. However, the computer-readable medium is not limited to devices such as storage device **210**. For example, the computer-readable medium may include a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, an EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave embodied in an electrical, electromagnetic, infrared, or optical signal, or any other medium from which a computer can read. Execution of the sequences of instructions contained in the main memory **206** causes the processor **204** to perform the process steps described below. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

Audit server **120** also includes a communication interface **218** coupled to the bus **202**. Communication interface **218** provides a two-way data communication as is known. For example, communication interface **218** may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface **218** may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. In the preferred embodiment the communication interface **218** is coupled to the network cable **102**. Wireless links may also be implemented. In any such implementation, communication interface **218** sends and receives electrical, electromagnetic or optical signals which carry digital data streams representing various types of information. Of particular note, the communications through interface **218** permits the transmission or receipt of audit trail information.

To assist in fully describing the present embodiment of audit reduction system the following terms are used with the following definitions. Note, however, that although a term may be herein defined, this does not necessarily exclude an established definition for the term if using the established definition is consistent with the purpose and scope of the present invention. "Audit trail information" includes (1) system audit trail records; (2) processing system log file data; and (3) processing system-maintained security state data. The audit trail information is received as native data and recorded "Authentication" entails associating a user with a system identifier. A "subject" is an active entity, generally in the form of a person, process, or device, that causes information to flow among objects or changes in the processing system state. An "object" is a passive entity that contains or receives information. Access to an object implies access to the information it contains.

An "audit agent" is a group of processes that handles the conversion of native audit files to XHAC data and transfers that data to the audit server **120**. "Audit processing" is the ability to read an input stream of data and to identify what type of data it is (i.e., BSM, C2, etc.) so that transformation may occur if needed. This is the input function of the agent. "Transformation" or "conversion" refers to the changing of data from one format (i.e., BSM) to another format (e.g., XHAC). This is the output function of the agent. The communication processing portion of the agent handles all of the communication logic between the agent and the manager as well as communication between the various processes that make up the agent.

With the above-definitions, the present embodiment may be understood as an auditing system, generally indicated at **300** in FIG. **3**. The auditing system **300** includes the previously mentioned audit agent, indicated at **310**, which resides on the audit server **120** (not shown in FIG. **3**). The audit agent **310** receives native audits from each of the nodes on the network **104**, **106**, **108**, **110**, **112**, **114**, and **118**. These audit trails include, for example, 1) system audit trails; 2) system log file data; and 3) data from third party applications and programs. The data from third party applications may use their own form of logging. As is known, the third party applications are used on many different types of processing systems. All of the information from these native audits is typically transient information which is subject to change based on the current activities on the processing system. As explained in greater detail below, the audit agent **310** collects the audits in the form of an audit trail, filters the audits, reduces the number of audits and normalizes some of the audits to a standard format for output to a security indications and warnings system (an intrusion and misuse detection engine).

The logical architecture in FIG. **3** illustrates an audit reduction workbench **320** which can be provided for communication with the audit agent **310**. The audit reduction workbench **320** can be the personal computer **112** depicted in FIG. **1**. The audit reduction workbench **320** can be used to build, maintain and send compiled audit reduction scripts to the audit agent **310**. The audit reduction workbench **320** can be used to build audit processing rules and to build anomaly detection rules. A configuration tool **330** can be provided which can communicate with the audit agent **310**. The configuration tool **330** can be the personal computer **112** in FIG. **1**. The configuration tool can be used to define audit agent configurations and define audit processing parameters. Workbench **320** and tool **330** can be combined to use a single computer. Configuration tool **330** sends configuration settings to an archive **340**. Audit agent **310** can reside on the audit server **120** or can reside on one or more computers in the network. As described below, audit agent **310** sends native audits to an on-line audit file storage device **350** associated with audit server **120** and sends normalized audits to an audit analysis database storage device **360** associated with audit server **120**. Audit agent **370** sends real time audits to a real time monitor **370** and real time alerts to a real time alert monitor **380**. Real time monitor **380** can be used to view audit alerts in near real time, view alerts and messages from other tools, view reports and activate other tools. Monitors **370** and **380** can be combined into a single computer or monitor. An audit review tool **390**, which can be a personal computer, receives configuration information from archive **340**, on-line audit files from on-line audit file storage device **350** and normalized audit information from audit analysis database storage device **360**. An ad hoc query tool **400**, which can be a personal computer, receives normalized audit information from audit analysis database storage device **360**. Stored queries can be defined and processed periodically. Ad hoc queries can be generated and executed immediately. An anomaly reporting tool **410** receives normalized audit information from audit analysis database storage device **360** and can forward this information to the alert monitor tool **380**. The anomaly reporting tool can generate alerts in near real time, replay audit scenarios and produce summary reports and statistics. An administrative tool **420** is associated with audit server **120** and may be in the form of a personal computer. The administrative tool **420** is used to define and maintain user descriptions/privileges, define and maintain group descriptions and define and maintain machine descriptions.

As depicted in FIG. **4**, the logical architecture of the audit agent **310** is depicted. Two mechanisms can be used to provide audit reduction code to the audit agent **310**. The first mechanism is that the code can be specified in a script **450**. The second mechanism is an audit reduction compiler **460**. Using either mechanism, instructions are inserted into address space within the audit agent **310**. There are five matrices that include information for the audit agent to use in dealing with incoming information. The first matrix is an environment parameters matrix **500** which provides information such as where to retrieve files, where the files are located in the directory structure, the naming convention, which operating system generated the files, and so forth. The second matrix is an identification matrix **510** which identifies each of the records in an incoming data stream based on information contained within each record. The third matrix is a transform matrix **520** which, once a record has been identified, can take one of two actions. The first action is direct transform meaning that the record is written into a different format and sent to the audit analysis database storage device **360**. The second action is to send the record to a reduction matrix **540** (fifth matrix) and reduce out the record. Advantageously, in the present invention, about ninety percent of the records are reduced out and only about ten percent of the records are transformed and further analysis is performed. The fourth matrix is a token matrix **530** which allows the audit agent **310** to find specific information within a record.

As previously mentioned, the audit agent **310** receives three types of native audits: 1) system audit trails **550**; 2) system log file data **560**; and 3) data from third party applications and programs **570**. The audit agent **310** can access shared libraries **340**. Normalized audits and native audits are sent by the audit agent **310** to storage devices **360**, **350**, respectively using a transport protocol. Status and control are also transmitted and received within audit reduction system **300** using a transport protocol.

The native audits are special purpose records that include information about system activity and transactions. These native audits include data written in a particular format. This formatted information may be called an audit record. Audit trails contain records showing who has accessed a computer system and what operations he or she has performed during a given period of time. Audit trails are useful both for maintaining security and recovering lost transactions. A plurality of records constitute a file. Data and records from on-line audit file storage device **350** and archive **340** can be used for audit system **300** to reconstitute actual events that occur within the processing system.

Refer now to FIG. **5** where a native audit record is illustrated. The native formats can be in one of two formats; either positionally based or contextually based. A contextually based record is, for example, two different business

cards from two different organizations where the information is positioned in different locations on the business card. On one business card the name may be in the middle of the card and on another business card, the name may be in the upper left hand comer. The context of the information, not the position, controls in a contextually based record. The best example of a contextually based format is syslog where the data can be free form. In this format the next field is determined by the previous field.

Syslog Example (Each field determines what should follow by the value in the field.)

1. VMUNIX: Reboot: This is a Reboot Message.
2. SendMail: Error reading SendMail config.
3. This is a plain text string.

For simplicity of explanation, the present invention is described in relation to the positionally based record depicted in FIG. **5**. A positionally based record includes information contained in particular positions in the record. The audit agent **310** has access to templates to recognize contextually based records and positionally based records and obtain information from either type of record. However, for ease of discussion, the processing of contextually based records is not further discussed herein. It should be appreciated that the record illustrated in FIG. **5** is simplistic and is only being used for ease of explanation. It should be further appreciated that many other positionally based records and contextually based records can be used in the invention.

A brief overview of how the audit agent **310** uses the information contained in an audit trail record is now provided with a more detailed explanation provided in the detailed flow diagrams (FIGS. **8–39**) and discussion thereof. A record is a stream of binary information. The record illustrated in FIG. **5** has an origin O which is the beginning of the record. The record comprises a plurality of fields. The first field (Field A) contains information concerning record length (RL). The second field (Field B) contains a primary discriminator (PID), for example 6, in the second field. The primary discriminator includes information on the type of record. There may be additional fields here, in this example, designated C, D, E, and perhaps other data thereafter.

For purposes of explanation, the previously described record having a PID of 4 has a format identifier (ID). For example, this record format may have an ID=5. All the records that are received having an ID=5 are going to be processed using the same set of instructions. The record is compared against a global format (see FIG. **5B**) for an ID=5. An immediate warning or alert message can be sent to a system administrator if certain criteria are met. The criteria might be that someone is trying to hack into the data stream. For example, if invalid data is in the incoming record stream, it may be an indication that someone is hacking into the data stream. Expert systems use this method for determining intrusions and misuses. The global format contains a set of instructions for all formats, including format 5. This way cross format rules can be supported. The global format (GF) includes information as to which output format (OF) that is associated with this input record. The output format can be an ASCII representation of the input record, or it can be a normalized record type, or it can be any desired output format.

After the record has been compared against the global format and an output format determined, then the record is compared against a set of tables called record entity tables (see FIG. **6**). As depicted in FIG. **6**, global items, such as format, record entities (control REC), and the like are always operated on first to provide for cross format operations. There are specific formats, for example, format 5 may be for system logins (syslog). Another format might be format **19** for SOLBSM. Another format is format 21 for SLOLMW. From the global format at **610**, the process continues to **620** in which each record entity is provided with a number. A matrix is provided at **630** for conversion rows. For example, REC record size, record type, date and user, and so forth. A matrix at **640** includes templates for matching records for logins, logouts and mounts. A matrix is provided at **650** which includes instructions for user equal route, record type equal login and date greater than a certain date such as October. A scenario matrix is provided at **660** which can be a chronological matrix, a bad logins matrix and a bad super-user scenario. At **670**, a template matrix is provided which might include a login template, bad login template and a bad super-user template. At **680**, a build row template is provided which might include a reduced PID build row template, reduce user build row template and a reduce record-type build row template. At **690**, an action template is provided which could either be reduce, transform or reduce. At **700**, various scenarios are created as instances or "copies" of the scenarios already activated. There are various scenarios that might be currently in progress such as a three bad logins scenario, three bad logins on JHW's computer, three bad logins on WAC's computer or a bad super-user (SU). There are various pointers denoted by dashed lines which are used based on the scenarios for the template matrix **670**, the build row template **680**, and the action template **690**. If a reduction is currently in progress at **692**, there is an elimination list of matrix. These lists where entries are created "on the fly" by successfully completed scenarios by the build rows. At **694**, there is an elimination matrix which includes certain elements which were previously audit records.

The audit agent **310** starts going through these record entities tables and tries to identify the record typically based on the primary discriminator. For example, in this simplistic view, if the primary discriminator includes a 6 (see FIG. **5A**), then a record entity corresponding to a record type=6 is activated. This identifies the record. There is an action associated with the identification of the record.

The action illustrated in FIG. **6** is a decision whether to reduce the record (throw the record out) from further consideration or normalize the record into a standardized format and then to save the record for further analysis by the intrusion and misuse detection engine. There are three types of reductions: simple-simple, simple-complex, and complex-complex. Simple-simple means one record is compared and interpreted against one template. Simple-complex means one record is compared and interpreted against many templates, but it takes only one record to move from a comparison phase state to an interpretation phase. Complex-complex means many records are compared and interpreted against many templates.

In a simple-simple reduction, the record is reviewed against a single record entity or template, and either there is a conversion, or the record is reduced out. If reduced, the record is not converted to another format but is sent to the on-line audit file storage device **350**. Then the next record is analyzed.

Reduction occurs in two phases: comparison (to see if something has happened) and interpretation (do something about it). To connect records that are somehow associated, the software uses defined fields as linkage fields. Linkage fields can include the process identifier (PID) (linkage or grouping by process), user identifier (UID) (linkage by user), etc. For example, the information contained in field C

could be the linkage field. All record-type=6 records are operated on through the linkage field. In this example, all the record-type=6 records that have a matching port ID or processing ID are matched for that record and then all the subsequent records that match that record are reduced out. For example, all records having a record-type=6 in field B and an 8 in field C (8 representing the matching part or processing ID).

In a complex-complex reduction, one record entity might find a match for a record-type=6 and if field C contains an 8, there is a whole series of records to match. In this example, the audit agent may search for another record that has a record-type=6, and so forth. Additional pattern matching could be required.

A simple-complex reduction is depicted in FIG. **7B** in which a record-type=6 in field B and a PID of 1234 in field E is compared against a template in the comparison phase. In the interpretation phase all records with a PID of 1234 are compared against an elimination list.

A complex-complex reduction is depicted in FIG. **7C** in which once a record having a record-type=6 in field B and a PID=1234 in field E is located in the comparison phase, all subsequent records having record-types equal to either 6 or 8 are compared against all records with a PID=1234 during the interpretation phase.

In FIG. **7D**, a flow diagram of the steps of FIG. **6** is illustrated. In step **610**, incoming audit records are received. At step **620**, an audit record is in the view and each of the audit records is matched against templates and elimination lists to determine if each audit record is to be reduced or further processed. If the audit record is marked for reduction at step **630** then, at step **640**, the record is copied for an in progress active scenario (in case scenario fails) as discussed in detail below. At step **650**, the audit record is either reduced or sent for conversion.

Actions can be taken depending on the matching based on the fact that the audit agent **310** failed or succeeded in finding a match. For example, a match may be expected such as in a financial transaction where a certificate is provided. If a certificate is provided, the records should be in a certain order or the fields in certain records should contain certain information, then the system may generate an alert that this certificate is not adhering to the standard methodology for a certificate. Another example is logins and logouts. For example, if record-type=6 in one record and then record-type=6 in another record for one computer on the network, this means this is a login (record-type=4) and this is a logout (record-type=6) and there is over twenty-four hours in between these two records, the audit agent **310** an alert may be provided because one machine can not be logged on for twenty-four hours, so that means that the user left their machine logged on. This is a potential security breach because a misuser could be using the logged on computer. In a complex-complex situation, a record may be matched against hundreds and hundreds of templates and different fields from each record compared against different templates. One match may cause additional matching operations to be performed.

An example of a simple-complex reduction includes successful logins on a SUN Solaris 2.5 system. With all audits enabled, the system will generate about 150 audit events—all considered normal activity. Using the simple-complex methodology, a scenario would become active at the receipt of a record with a record-type of 4 (Example) subsequently an elimination list entry is build to reduce out the 149 subsequent audits based on the PID linkage field. The initial record generates the login output record. An example of a complex-complex scenario includes the tracking of user operations to ensure that the sequence is completed correctly. In this application of the technology it is necessary to step through the sequence of records while in the comparison phase to identify if the user actions occurred correctly. Some records are necessary to the completion of the scenario but some intermixed (within the linkage constraints) records may be ignored. All comparisons in the template list must succeed prior to switching to the interpretive phase.

Specifically, the analysis of a chronology (cron) job scenario requires a complex-complex definition. First, the system must detect a login event. Then a number of activities occur associated with the system starting the job, i.e., the generation of a shell, the reading of the even tab file.

In summary, the audit agent **310** performs a first line analysis and reduces the audit records to be output and then outputs the audit records. The audit agent can also wait for additional records. The analyzed, reduced and transformed data is then output to audit analysis database storage device **360**. This data can then be further analyzed by the intrusion and misuse detection engine disclosed in a patent application entitled "Method and System for Detecting Intrusion into and Misuse of a Data Processing System" assigned to the instant assignee and filed on even date herewith and incorporated by reference into this specification in its entirety.

With this brief overview in mind, the details of the present invention can be understood with reference to FIG. **8** where a top-level flowchart is depicted according to the present invention. The flowcharts illustrated in FIGS. **9–39** are details of the steps in FIG. **8**. At step **800** in FIG. **8** the process is started. At step **805**, the process is initialized. The initializing step is illustrated in greater detail in FIG. **9**. At step **910**, in FIG. **9**, tables are built, by setting up, for each format (positional and contextual), the five matrices **500**, **510**, **520**, **530**, **540** discussed above with respect to FIG. **4**. At step **920**, communications are set up by initializing any sockets, etc. All the matrices are populated within the address space, so now the instructions are available for the audit agent **310** to start processing. At step **930**, the initializing step **805** is complete and the process proceeds to the sets up audit processing step **810** in FIG. **8**.

The set up audit processing step **810** is illustrated in detail in FIG. **10** and at step **1000** the process is started. At step **1005**, the audit agent **310** decides whether to wait for additional format types. For example, there may be one or more audit types that are positionally based, and one or more audit types that are contextually based. If there are no more types to wait for, at step **1010**, the set up audit processing step **810** is complete. At step **1015**, the audit agent parses an intermediate audit reduction (AR) file, and builds tables. A file is created that allows the audit agent **310** to make temporary changes to the file without making permanent changes. At step **1020**, the configuration of the record is read and the type of configuration is determined, based on the environment constraints **500**. The environment constraints **500** is used to determine where the record comes from. At step **1025**, a parse reduction file can be set up although it is not necessary. At step **1030**, a scan directory step provides information regarding the location of the audit files. For example, the audit files may be sent by the audit agent **310** in real time or may be extracted from the on-line audit files database **350**. At step **1035**, it is determined whether any audit files exist. If there are audit files, then at step **1040**, the file table is updated. Otherwise at step **1045** the system returns to step **1005**. When all of the types of records have been set up, the set up processing step is completed at step

1010. This set up audit processing step 810 is performed for all the different format types that are going to be processed.

Returning to FIG. 8 at step 815, there is check for a terminate signal from the server or alert monitor tool 380 to shut down. At step 820, there is check for any change signals to change the current configuration. A change signal would be sent from the anomaly reporting tool 410. At step 825, if there is a change request, changes are made to the tables and so forth.

At step 835, the process audit types step is begun. Step 835 is conducted sequentially for each audit type. The process audit types step 835 is illustrated in detail in FIG. 11 and at step 1100 the process is started. At step 1105, the audit type is checked. At step 1110, if there are no files to process then the process audit types step is complete. At step 1115, it is determined whether there are any files to process. At step 1120, if there are files to process then the files are processed and after the files are processed the process continues to step 1135. At step 1125, audit files are selected based on criteria. The criteria includes a time range as well as a naming convention for the current source. At step 1130, once files have been selected the process returns to step 1115. If there are no new files, then the process proceeds to step 1135. Step 1135 returns the process to step 1105.

The process file step 1120 is illustrated in greater detail in FIG. 12. At step 1200, the process file step 1120 is started. At step 1205, the programmer use file location is checked against a type equals file location to see if this is an open file. At step 1210, the file is checked to see if has been previously processed, and if any new data has been added to this file. At step 1215, the amount of process time for the file is checked and if it exceeds a predetermined amount of time, then at step 1220, the process is completed. At step 1225, the file location is checked against file size. If the file location equals file size no new data has been added to that particular record and the process proceeds to step 1230. If the file location does not equal the file size, then at step 1235, this means that the operating system has only written a partial record. At step 1240, if the file size minus the file location size is less than a buffer size then additional records can be added to the buffer until, at step 1245, there is a full buffer. The buffer includes a plurality of records forming a portion of a file. At step 1250, the file buffer size is read and then at step 1255 the process buffer step 1255 is started. Step 1255 loops back to step 1230 and then to step 1215 until all of the records have been read.

The process buffer step 1255 is illustrated in greater detail in FIG. 13. At step 1300 the process buffer step is started. At step 1305, a pointer is set to the beginning of the buffer. The pointer advances as it goes through the buffer and identifies records. At step 1310 the process waits for more records. If there are no more records, at step 1315 the file location is equal to buffer location pointer. At step 1320 the file table is updated and at step 1325 the process waits until the next cycle.

At step 1330, if there are additional records, the additional records are identified and the record size determined. At step 1335, if there are no more records (end of the audit file) the process proceeds on to step 1360 where the file is marked, and the buffer location pointer is set to zero at step 1365. If it is not the end of the audit file, at step 1335, the process checks for a file header at step 1340. If there is no file header, at step 1340, it must be determined, at step 1345, whether to perform further processing, meaning whether to convert the record to a standardized format or to reduce the record out by advancing the buffer at step 1355. At step 1350, the conversion is performed which is discussed in greater detail below.

FIG. 14 is the locate event type flow chart. At step 1400 the process is started. At step 1410 the primary discriminator is located. At step 1420 the ID pointer is set to a primary discriminator offset. The primary discriminator (PD) performs the grossest level of filtering. It provides for a rapid narrowing of potential rules for the process to check against. Similar in concept to a sort, i.e., if the PD is "X" and only four rules apply to a record with a PD of "X" the processing can be saved of rules for PDs of "Y", "Z", and "A" because they can't apply. The PD offset tells the process where to find the PD within the record (it varies depending on the format, and in certain cases doesn't exist). At step 1430, the match function is performed and at step 1440, the process is complete.

The match function step 1430 is illustrated in greater detail in FIG. 15. At step 1500 the match function step is started. At step 1505, the match index is set to zero meaning that the process starts with the first match record. Records that are matched need to be checked again for additional matches. At step 1505, the process is set to the beginning of a match table. At step 1510, the process looks for the end of the match table and checks for the end of file. As previously mentioned above with respect to step 1410, the primary discriminator has already been identified. The match function is a table in which is stored the particular templates that a record should be compared against based on the primary discriminator. This is the same as the records entity templates of FIG. 5). Once the process reaches the end of file, the process exits at step 1515. If the process is not at the end of the match table, then processing is continued at step 1518. At step 1520, there is a binary comparison of two memory locations to determine if there is a match between the two values. This can be two values in two fields or one value in the record and the other value in the template. If there is not a match at step 1520, then the pointer is advanced in the match table and the process goes on to the next match record at step 1565. At step 1520, if there is a match, at step 1525 the match command is set to zero. Match commands are the "holes" in the templates, i.e., how to conduct the match. A match command table is another table that is a sub grouping of instructions from the original match table. At step 1530, using the match command table, it is determined if the pointer is at the end of the table. If there are additional templates to compare then the process proceeds to step 1535. If the match is equal to zero at step 1560 then there are no more matches. There was at least previous one match for the primary discriminator but not using any of the additional templates. If there are additional matches, at step 1535 then the process proceeds to step 1540, and more match commands are provided if there is a positive match result at step 1545. The match result is produced at step 1545 and at step 1550 the pointer is set to continual and then the process exits out at step 1530. If the process is not at the end of the match command, at step 1530, the absolute value of the match is taken which is always expected to be positive if it is successful. Commands can be successful if the command fails. The audit reduction agent 310 requires a positive return from each command processed, otherwise the match fails. However, in certain cases, negative logic is used (i.e., if something is not equal or fails to match). The expected state is present so that when a failure is expected and one is obtained, this is considered success. The match command is incremented at step 1560 and this loop is continued until all the match commands for this match record have been checked. Depending on the match obtained at step 1560, then at step 1565 it is determined whether to continue with the additional match records or exit.

The process command step **1535** is illustrated in greater detail in FIG. **16**. At step **1600** the process command step is started. At step, **1605** an arbitrary command is checked to see if it is greater than **95**. If the command is greater than **95**, then it is considered conditional. The process is checking here to see if there is an exit or fault command. At step **1610**, the process checks to see if the command is less than **100** and an X flag is set at step **1615**. At step **1620** there are cases when it is desired that the result obtained returns false. At steps **1605–1615**, if the command is greater than **95**, the result is expected to be false. So a true is returned in that case. Processing can not continue unless a true is returned. These arbitrary values have to pre-set so that the process is successful in recognizing a false state. At step **1620** the case instructs the audit agent **310** to operate on any particular field within the data. This moves the pointer in the data to a new location. At step **1625**, the match pointer is set equal to the buffer location pointer plus an offset to perform a comparison. At step **1635** a binary comparison occurs with the pointer being in the table. If the X flag is set in step **1615**, then the X flag says that the process is supposed exit if this is true. A primary discriminator has been identified so the process has the functionality to create a program about how to continue to identify this record. The record or data can be compared to other data, pointers can be advanced to look for other values within the record and the process command takes what is written in the command table and executes the commands in the command table.

Step **1625** provides the ability to advance the pointer within the data and look at other fields besides the primary discriminator. The binary compare step **1635** can be used to compare a field of data against other fields of data in other databases. For example, the data could be contained within other tables, memory or even contained in interface **118**. Step **1640** is a determination to set the X flag based on a set of conditions. The audit education agent **310** requires a positive return from each command processed, otherwise the match fails. However, in certain cases, negative logic is used (i.e., if something is not equal or fails to match). The expected state is present so that when a failure is expected, and one is obtained, this is considered success (X=exit on true). If certain conditions are met at step **1645**, the process exits. If the conditions are not met at step **1650**, a retrieval is obtained which equals the results plus a command which is a numeric value (command value numeric). At step **1655**, the process exits. Step **1630** is an error handler for an unknown value.

The perform conversion step **1350** in FIG. **13** is illustrated in greater detail in FIG. **17**. At step **1700**, the process is started and at step **1705**, the conversion index is set to **0**. At step **1710**, conversion is performed until the end of table is reached or until a match is found. If no match is found, then at step **1715**, the native audit is reduced. At step **1720**, it is determined whether a reduction has been performed. If no reduction has been performed at step **1725**, the record is written to native audit on-line audit file **350**. If the reduction has occurred at step **1720**, then the process is complete at step **1730**. At step **1710**, if the end of file has not been reached, at step **1735**, a conversion row is started. There are 1 to N number of conversion rows. There is an input conversion row and an output conversion row which tells the audit agent **310** how to convert the data and where to put the converted data. At step **1740**, if the length of the converted row is equal to −1, then the audit agent **310** does not know what the length of the converted record is. At step **1745**, a delimiter is determined for that record. If the conversion row is equal to −1, then at step **1755**, the token entry is located. The token entry delimits the field definition. For example, it might be a colon, or some other type of data that provides information that the end of field has been reached. If there is no token as determined at step **1760**, then at step **1765** and **1770**, the length of the audit record is determined. At step **1775**, if the token is null, then the string length is determined. Using either step **1770** or **1775**, the string length can be determined and the amount of data limited. In either event, at step **1750**, other codes are executed to perform specific conversion on the data. For example, there is a registered call back function which gets executed because the data is being converted from an input format to an output format. The process loops back to step **1710** until all the conversion rows and data have been converted. All the data will be mapped into the conversion table to be converted.

Reduction step **1715** is illustrated in greater detail in FIG. **18**. At step **1800**, the reduction step is started. At step **1805** there is a determination step to determine whether the record is in an elimination list **1810** by comparing the record against the elimination list **1810**. The record is matched against every entry in the elimination list **1810**. If the record matches any of the entries, the record will be reduced out. At the beginning, there are no active scenarios in the elimination list **1810**. There are three lists of scenarios. The first is the list of possible scenarios that can happen as defined in the memory by the AR programmer. The second is a list of active scenarios in memory. This could include scenarios; for example, SC1–SC6, as depicted below. Because certain events have occurred in the audit stream (chronological audit data in order of generation), a scenario is said to be "in progress" or "active". This could include scenarios SC3, SC3, SC1, and SC6 as depicted below. This means the audit agent **310** will look for subsequent audit records (as matched against templates) as opposed to the already reviewed audit records.

Life cycle of a scenario:

Can happen Possible Scenarios (as defined by the ARL Programmer	Are happening In Progress Scenarios (Comparison Stage) Have Been Activated by Events	Have happened Elimination List Generated by Successful Scenarios to Reduce Out Additional Records
SC1	SC3	SC3 PID = 21
SC2	SC3	SC3 PID = 22
SC3	SC1	SC3 PID = 23
SC4	SC6	SC3A PID = 100
SC5		SC3A PID = 101
SC6		SC3A PID = 102
Etc.		SC1
		Etc.

At step **1815**, there is a determination as to whether there are any currently active scenarios. A scenario includes two or more identifiers within a record. When the audit agent first begins, there are no active scenarios in the elimination list but scenarios will be added as the process continues as discussed in detailed below. At step **1815**, it is determined whether there are any active scenarios. An active scenario list is provided at **1835**. How the active scenario list is created is discussed below. For the purposes of FIG. **18**, it must be assumed that there are some active scenarios in list **1835**. Step **1830** determines whether the record matches any current scenarios in the active scenario list. If there are no matches, then at step **1840**, it should be determined whether to create new scenario parameters. This means that a new scenario is being added to the active scenario list **1835**. If

step **1830** is positive and matches located, then the record is compared against other templates. The active scenario list **1835** means that the scenarios are currently waiting for a record. For example, there may be thirty-two active scenarios, but the active scenario is waiting for a particular record. At step **1825**, the reduction process is stopped.

The record in elimination list step **1805** is illustrated in greater detail in FIG. **19**. At step **1900**, the process is started. At step **1905**, the reduction list is selected to search by format. At step **1910**, it is determined whether a match has occurred against a list of reduction entries. If no match has occurred, then the process is completed at step **1915**. If the process is continuing to perform matches, then at step **1920**, timers are checked. At step **1925**, the ordinality is validated. Most comparisons and templates work on ordinality of "fields" (areas of data within a record). Step **1925** ensures pointers are at the appropriate location based on ordinality. At step **1930**, the audit records are checked for completeness. At step **1935**, it is determined whether the audit data matches the reduction list data. If the audit data does not match the reduction list data, then the process returns to step **1910**. If the audit data does match the reduction list data, then at **1940**, the reduction list match is processed.

The processed reduction list match step **1940** is illustrated in greater detail in FIG. **20**. At step **2000**, the process is started. At step **2005**, it is determined whether the reduction sequence could be decremented. A grouping of elimination list records which are associated—decrementing refers to the popping of a elim list record from the associated record "stack", i.e., making the next record in the sequence active. If the reduction sequence can be decremented, then at step **2010**, the reduction sequence is decremented. If the reduction sequence can not be decremented, then at step **2015**, it is determined whether the matched reduction entries life span is temporary. If the matched reduction entries life span is temporary, then at step **2020**, the matched list entry is removed. At step **2025**, the process is complete.

Determination step **2005** is illustrated in greater detail in FIG. **21**. At step **2100**, the process is started. At step **2105**, the previous list entry is compared against criteria. Criteria can be any values, ranges, etc., based on scenario instructions. At step **2110**, the next list entry is compared against criteria. At step **2115**, it must be determined whether the previous and next records meet the criteria. If the previous and next records meet the criteria, then at step **2120**, the sequence may be decremented. And at step **2125**, the process is complete. If the previous and next records do not meet the criteria, then at step **2130**, the sequence may not be decremented and process is completed at step **2125**.

The compare previous list entry against criteria step **2105** is illustrated in greater detail in FIG. **22**. At step **2200**, the process is started. At step **2205**, it is determined whether the previous record is null. If the previous record is null, then at step **2210**, the previous record meets the criteria. If the previous record was not null, then at step **2215**, it must be determined whether the matched IDs are equal. If the matched IDs are not equal, then the previous record meets the criteria at step **2210**. If the previous record and/or row, then the previous record meets the criteria at step **2210**. If the previous record and/or row is not determined, then at step **2225**, it must be determined whether the previous record has a sequence of 1. If the previous record does not have a sequence of 1, then the previous record meets the criteria at step **2210**. If the previous record does have a sequence of 1, then at step **2230**, the previous record does not meet the criteria. From either step **2210** or **2230**, the process proceeds to step **2235**, where this portion of the process is complete.

The compare next list entry against criteria step **2110** is illustrated in greater detail in FIG. **23**. At step **2305**, the process is started. At step **2310**, it is determined whether the next record is null. If the next record is null, then at step **2315**, the next record meets the criteria. If the next record is not null, it must be determined at step **2320**, whether the matched IDs are not equal. If the matched IDs are not equal, then proceed to step **2315**. If the match IDs are not equal, then at step **2325**, it must be determined, is it the next record and/or row. If it is, then proceed to step **2315**. If it is not, then step **2330** must be determined as the next record life span temporary. If it is, then proceed to step **2315**. If it is not, then at step **2335**, it must be determined whether the next record sequence is greater than 1. If it is, then proceed to step **2315**. If it is not greater than 1, then at step **2340**, it must be determined whether the next record sequence is equal to 1. If it is not, then the next record meets the criteria at step **2315**. If it is, then at step **2345**, the next record does not meet the criteria. From either step **2315** or **2345**, the process proceeds to step **2350** where the process is complete.

The decrement reduction sequence step **2120** is illustrated in greater detail in FIG. **24**. At step **2400**, the process is started. At step **2405**, it must be determined whether there are entries in the list and the matched IDs that are equal. If there are none, then the process proceeds to step **2410** where the process is complete. If there are entries in the list and match IDs that are equal, then at step **2415** it must be determined whether the sequence is greater than 1. If the sequence is not greater than 1 then the process returns to step **2405**. If the sequence is greater than 1, then at step **2420**, the entries sequence number is decremented by 1. Then the process returns to step **2405**.

Steps **1845** and **1855** both proceed to step **2500** as illustrated in FIG. **25**. FIG. **25** depicts the process of a complex second stage interpretation. At step **2505**, the next template record is read in. At step **2510**, it is determined whether the end of the template list has been reached. At step **2510**, if the end of the template list has not been reached, then it is decided whether to add this elimination scenario (plurality of templates) to the elimination list **1810**. If the answer to step **2515** is yes, then at step **2520**, the plurality of templates which make up a scenario are added to the elimination list. If the answer to step **2515** is no, then the process stops at step **2525**. At step **2530**, because the scenario did not generate any matching templates, the scenario is removed at step **2530**. At step **2535**, the scenario is added to a post-mortem list thereby taking the scenario off of the active scenario list **1835**. At step **2540**, it is determined whether there are more scenarios to consider. If there are more scenarios to consider, the process proceeds to step **2525** and stops. If there are no more active scenarios, then at step **2545**, the process continues to step **2525** and the process is stopped.

The removed scenario step **2530** is illustrated in greater detail in FIG. **26**. At step **2600**, the removed scenario step is started. At step **2605**, the counter associated with the open scenario list is adjusted by one. The scenario is deleted from the active scenario list **1835**. At step **2615**, the open scenario list is updated and at step **2620**, the removed scenario step is stopped.

The add to elimination list step **2520** is illustrated in greater detail in FIG. **27**. At step **2700**, the add to elimination list step is started. At step **2705**, the scenario is added to the elimination list **1810**. At step **2715**, the open elimination list counter is incremented. At step **2720**, it is determined whether the open elimination list has reached a maximum size and used up all the memory. If the elimination list **1810**

has reached a maximum size, then a scenario is deleted at step **2730** and the open elimination list is revised at step **2735**. From either step **2720** or **2735**, the process is then stopped at step **2725**.

The add scenario step **2705** is illustrated in greater detail in FIG. **28**. At step **2800**, the process is started. At step **2805**, the audit agent **310** gets the action (see FIG. **8**). At step **2810**, the simple-complex flag is retrieved. At step **2815**, it must be determined whether there are still elements in the scenario list. If there are not, then the process proceeds to step **2820** and is complete. If there are still elements in the scenario list then the action is processed at step **2825**.

The action step **2825** is illustrated in greater detail in FIG. **29**. At step **2900**, the process is started. At step **2905**, template information is retrieved. At step **2910**, it must be determined whether the action is to reduce. If the action is to reduce, then at step **2915**, a new reduction list entry is set up. If the action is not to reduce, then at step **2920** it must be determined whether there are any more build rows for this template. If there are not, then at step **2925**, the process is complete. If there are more build rows for this template, then at step **2930**, the build row is processed. At step **2935**, it must be determined whether the action is to reduce. If the action is to reduce, then a new reduction list entry is set up. If the action is not to reduce, then the process returns to step **2920**.

The set up new reduction list entry step **2940** is illustrated in greater detail in FIG. **30**. At step **3000**, the process is started. At step **3010**, space is allocated for a new reduction entry. At step **3020**, the interpretation time is set to live. At step **3030**, the match _ID is set and at **3040**, the process is complete.

The set interpretation time to live (ittl), step **3020**, is illustrated in greater detail in FIG. **31**. At step **3100**, the process is started. At step **3105**, it is determined whether this is a new scenario. If it is not a new scenario, then at step **3110**, the entries time to live is listed as equal to the previous list entries ittl. At step **3115**, the process is complete. At step **3105**, if this was a new scenario, then at step **3120**, it must be determined whether the scenarios ittl is valid. If the scenarios ittl is valid, then at step **3130**, the entries time to live is listed as equal to the scenarios ittl. If at step **3120**, the scenarios ittl is not valid, then at step **3125**, the entries time to live is listed as equal to 30 seconds. From either steps **3125** or **3130** the process is complete at step **3115**.

The build row step **2930** is illustrated in greater detail in FIG. **32**. At step **3200**, the process is started. At step **3205**, the starting instructions and number of instructions are retrieved. At step **3210**, the build rows instructions are processed. At step **3215**, it is determined whether the process instructions were successful. If the process instructions were not successful, then a return value is listed as false at step **3220**. If the process instructions step **3215** was successful, then at step **3225**, it is determined what was the action to reduce. If the action was to reduce at step **3225**, then at step **3230**, the reduction parameters are checked. At step **3235**, the entry is added to the reduction list. At step **3240**, the process is complete. At step **3225**, if the action was not to reduce, then at step **3245**, it must be determined whether the advance record flag was set. If the advance record flag was not set, then the process is completed at step **3240**. If the advance record flag was set, then at step **3250**, the new output record is posted. At step **3255**, the output record is free and at step **3240**, the process is complete.

The process to build rows instruction step **3210** is illustrated in greater detail in FIG. **33**. At step **3300**, the process is started. At step **3305**, spaces are allocated in the memory for the primary identifier (PID). At step **3310**, the list entries sequence field is set. At step **3315**, the list entries ordinality field is set. At step **3320**, a pointer is positioned to the primary identifier in the audit record. At step **3325**, the primary identifier is stored from the audit record to the list entries. At step **3330**, the list entries primary discriminator length field is set. At step **3335**, the data type of the primary discriminator and list entry is set. At step **3340**, the list entries operations field is set. At step **3345**, the list entries life span field is set. At step **3350** the process is complete.

The active scenario list step **1845**, is illustrated in greater detail in FIG. **34**. At step **3400**, the process is started. At step **3405**, an element is added to the active scenario list **1835**. At step **3415**, open scenarios are added. At step **3420**, the process is complete.

The reread parameters step **830** is illustrated in greater detail in FIG. **35**. At step **3500**, the process is started. At step **3505**, the process waits for more types of audit data. If more data types are received, at step **3510**, there is a change the audit reduction file. At step **3515**, the intermediate audit reduction file and tables are parsed. If there is no change in the audit reduction at step **3510**, then the configuration is changed at step **3520**. If the configuration is changed at step **3520**, then at step **3525**, the configuration type is read. From either step **3515** or **3525**, the process continues to step **3540** and then loops back to step **3505**. When all of the audit types have been completed, then the process is complete at step **3545**.

The package audit data step **840** is illustrated in greater detail in FIG. **36**. At step **3600**, the process is started. At step **3605**, the packet built is equal no. At step **3610**, the process waits for more types of audit data. At step **3615**, if the data is being received in bulk, it must be determined what type of data is being received. If it is bulk data then it must be packetized at step **3620** and if the packet is not completed, then at step **3625**, the unpacketed data is sent to a buffer and then to step **3640**. After the packet is complete or if the data is not a bulk type data, then step **3630**, AITP (Audit Information Transfer Protocol) packets are built. At step **3635**, a signal is sent that the packet is built and the process proceeds to step **3640**. After all of the data types have been received in step **3610**, then the process proceeds to step **3645** where the return packet is built and at step **3650** the process is complete.

The send data step **850** of FIG. **8** is illustrated in greater detail in FIG. **37**. At step **3700**, the process is started. At step **3705**, a message is sent that the packet has not been sent. At step **3710**, the process waits for more native audit types. At step **3715**, it determines whether the data is being received in bulk. If the data is being received in bulk, then at step **3720**, it determines the type and the amount of time to process the bulk data. At step **3725**, it must be determined whether the time has been exceeded to process the bulk data. If the time has not been exceeded, then the process proceeds to step **3745** and back to step **3710**. If the time has been exceeded at step **3725**, then at step **3730**, then an AITP packet. At step **3735**, the time sent is reset. At step **3740**, a signal is sent that the packet has been sent. The process then proceeds back to step **3710**. From step **3710**, the process proceeds to step **3750** where a return packet is built and at step **3755**, the process is complete.

FIG. **38** is directed to a configuration change for the audit reduction agent **310**. At step **3800**, the process is started. At step **3805**, it is determined whether a configuration change command has been received. At step **3810**, if a configuration change command signal has been received, then the configuration is changed at **3810**. After the configuration change is posted at step **3810**, the process is completed at step **3830**.

If there has been no configuration change at step **3805**, then at step **3815**, it is checked to determine whether there has been an audit reduction change. If there has been an audit reduction change then at step **3820**, the audit reduction change is posted and the process is completed at step **3830**. If there has been no audit reduction change at step **3815** then at step **3825** the process is restarted. At step **3835**, a stat file is ridden. At step **3840**, the stat file is initialized. At step **3845**, processing is set up. From either steps **3825** or **3845**, the process proceeds to step **3830** and is complete.

The right stat file step **3835** is illustrated in greater detail in FIG. **39**. At step **3900** the process is started. At step **3905**, the process waits for more data types. If more data types are to be received, then at step **3910**, the type of data configuration is written. At step **3915**, the current AR format is written. At step **3920**, the current AR reduction type is written and the process loops back to step **3905**. If there are no more data types to be received, then at step **3925**, the stat file is written and at step **3930**, the process is complete.

It should now be apparent from the foregoing detailed description that a method of analyzing native audit data has been described. Advantageously, the method compares an audit record against one or more templates and against one or more audit records to determine whether the audit record might be indicative of a potential security threat. If the audit record is not considered to present a potential security threat, then the audit record is reduced. If the audit record or records represent a potential security threat, then the audit record or records are normalized to a standardized format for further evaluation by a misuse and intrusion detection engine.

It will be readily seen by one of ordinary skill in the art that the present invention fulfills all of the objects set forth above. After reading the foregoing specification, one of ordinary skill will be able to affect various changes, substitutions of equivalents and various other aspects of the invention as broadly disclosed herein. It is therefore intended that the protection granted hereon be limited only by the definition contained in the appended claims and equivalents thereof.

What is claimed is:

1. A method of reducing the volume of native audits received from at least one operating system, each of the native audits being in a particular format, comprising:

identifying the particular format for each of the received native audits;

comparing each of the received identified native audits against at least one template and determining if each of the native audits matches at least one template; and

eliminating all of the matched native audits before a misuse and intrusion detection engine further analyzes the native audits.

2. The method of claim **1**, further comprising transforming selected ones of the native audits to a standardized output.

3. The method of claim **2**, further comprising outputting the transformed audits to a transformed audit storage device.

4. The method of claim **2**, comprising associating each of the native audits with an output template for transforming the native audit to the standardized output format.

5. The method of claim **1**, wherein the native audits comprise at least one of system audit trails; system log file data; and data from third party applications and programs.

6. The method of claim **1**, comprising associating each of the native audits with a global format template, each of the global format templates including a listing of additional templates against which each of the associated native audits should be compared.

7. The method of claim **1**, further comprising outputting the native audits to a native audit storage device.

8. The method of claim **1**, comprising comparing one of the received native audits against a single template and either reducing the native audit and sending the native audit to a native audit storage device or transforming the native audit and sending the transformed audit to the transformed audit storage device.

9. The method of claim **1**, comprising comparing one of the received native audits against a plurality of templates and either reducing the native audit and sending the native audit to a native audit storage device or transforming the native audit and sending the transformed audit to the transformed audit storage device.

10. The method of claim **1**, comprising comparing a plurality of the received native audits against a plurality of templates and either reducing the plurality of native audits and sending the plurality of native audits to a native audit storage device or transforming the plurality of native audits and sending the plurality of transformed audits to the transformed audit storage device.

11. The method of claim **10**, wherein the native audits are in one of a contextual format or a positional format.

12. The method of claim **1**, comprising setting up at least one possible scenario and comparing the received native audits against the at least one possible scenario and activating the at least one possible scenario to become an in-progress scenario.

13. The method of claim **12**, comprising comparing subsequently received native audits against the in-progress scenario, a successfully completed scenario becoming an elimination list.

14. The method of claim **13**, comprising comparing each of the received native audits against the elimination list and if the received native audit matches a record on the elimination list, reducing the matched native audit.

15. The method of claim **1**, wherein each template describes a weighting system using barriers and boundaries so that a potential misuse or intrusion can be detected before it occurs.

16. An article of manufacture which includes instructions for reducing the volume of native audits received from at least one operating system, each of the native audits being in a particular format, wherein the instructions are on a computer readable medium to be executed by a computer system, comprising:

identifying the particular format for each of the received native audits;

compare each of the received identified native audits against at least one template and determine if each of the native audits matches at least one template; and

eliminate all of the matched native audits before a misuse and intrusion detection engine further analyzes the native audits.

17. The article of claim **16**, wherein each template describes a weighting system using barriers and boundaries so that a potential misuse or intrusion can be detected before it occurs.

18. A computer architecture for reducing the volume of native audits received from at least one operating system, each of the native audits being of particular format, comprising:

identifying means for identifying a particular format for each of the received native audits;

comparing means for comparing each of the received identified native audits against at least one template and

determining if each of the native audits matches at least one template; and

eliminating means for eliminating of the native audits before a misuse and intrusion detection engine further analyzes the native audits.

19. The computer architecture of claim **18**, wherein each template describes a weighting system using barriers and boundaries so that a potential misuse or intrusion can be detected before it occurs.

20. A computer system, comprising:

a processor; and

a memory coupled to said processor, the memory having stored therein sequences of instructions for reducing the volume of native audits received from at least one operating system, each of the native audits being in a particular format, the instructions which, when

executed by said processor, causes the processor to perform the steps of:

identifying a particular format for each of the received identified native audits;

compare each of the received identified native audits against at least one template and determine if each of the native audits matches at least one template; and

eliminate all of the matched audits before a misuse and intrusion detection engine further analyzes the native audits.

21. The computer system of claim **20**, wherein each template describes a weighting system using barriers and boundaries so that a potential misuse or intrusion can be detected before it occurs.

* * * * *

EXHIBIT D

US 20020174093A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2002/0174093 A1**

Casati et al. (43) **Pub. Date: Nov. 21, 2002**

(54) **METHOD OF IDENTIFYING AND ANALYZING BUSINESS PROCESSES FROM WORKFLOW AUDIT LOGS**

(76) Inventors: **Fabio Casati**, Palo Alto, CA (US); **Ming-Chien Shan**, Saratoga, CA (US); **Li-Jie Jin**, Mountain View, CA (US); **Umeshwar Dayal**, Saratoga, CA (US); **Daniela Grigori**, Nancy, CA (US); **Angela Bonifati**, Milano (IT)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: **09/860,230**

(22) Filed: **May 17, 2001**

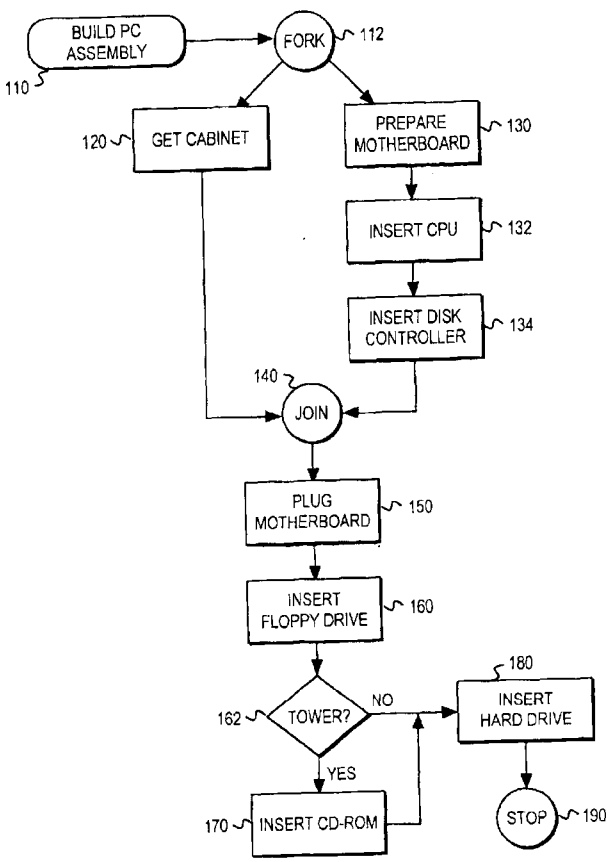
Publication Classification

(51) **Int. Cl.**[7] **G06F 7/00**

(52) **U.S. Cl.** **707/1**

(57) **ABSTRACT**

A method of identifying and analyzing business processes includes the step of populating a data warehouse database with data from a plurality of sources including an audit log. The audit log stores information from a plurality of instantiations of a defined process. The data is then analyzed to predict an outcome of a subsequent instance of the process. Data mining techniques such as pattern recognition are applied to the data warehouse data to identify specific patterns of execution. Once the patterns have been identified, the outcome of a subsequent instance of the process can be predicted at nodes other than just the start node. The probability of completion information can be used to modify resource assignments, execution paths, process definitions, activity priority, or resource assignment criteria in subsequent invocations of the defined process.



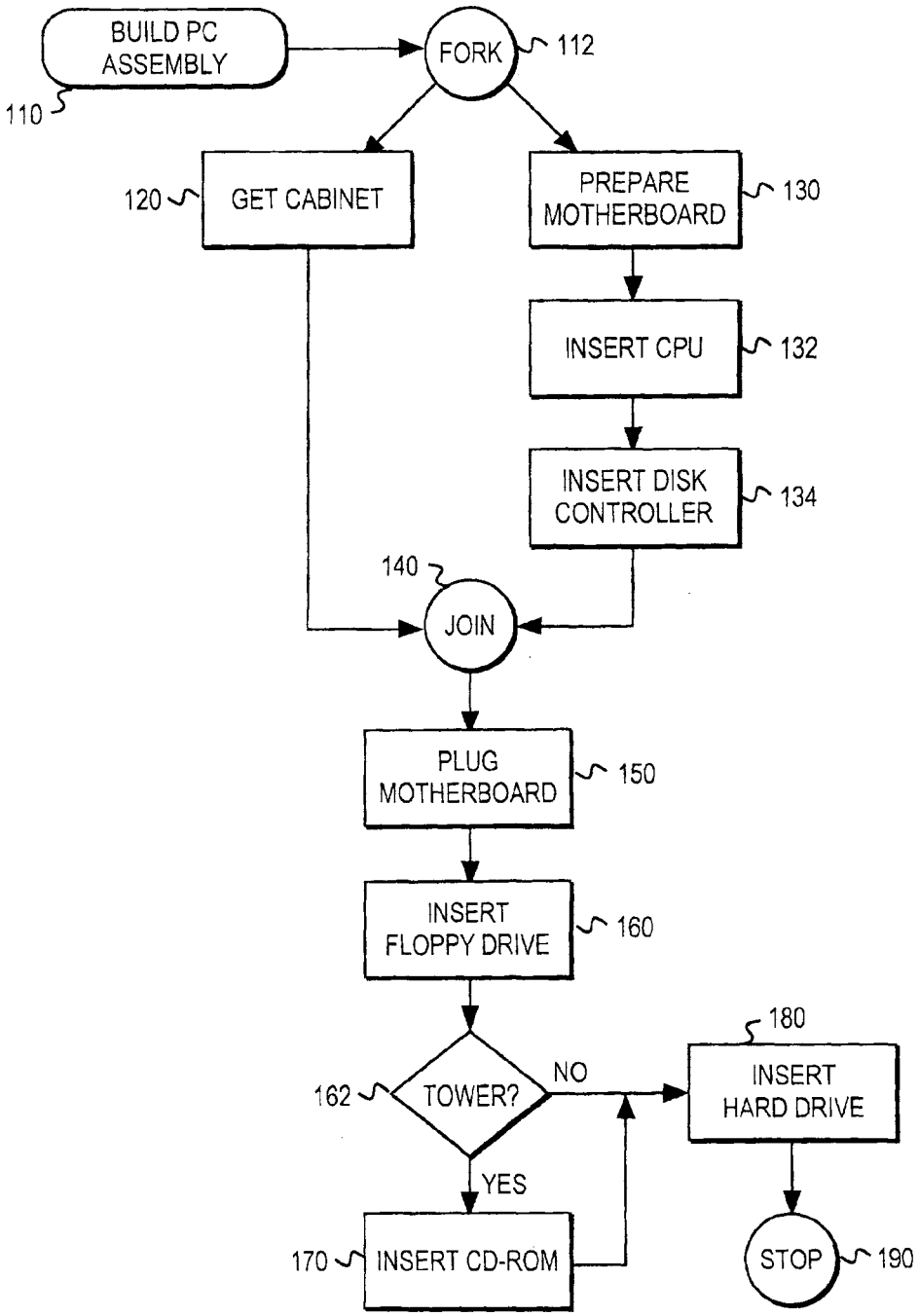

BUILD PC ASSEMBLY
110
FORK
112
GET CABINET
120
PREPARE MOTHERBOARD
130
INSERT CPU
132
INSERT DISK CONTROLLER
134
140
JOIN
PLUG MOTHERBOARD
150
INSERT FLOPPY DRIVE
160
180
TOWER?
162
NO
INSERT HARD DRIVE
YES
INSERT CD-ROM
170
STOP
190

FIG. 1

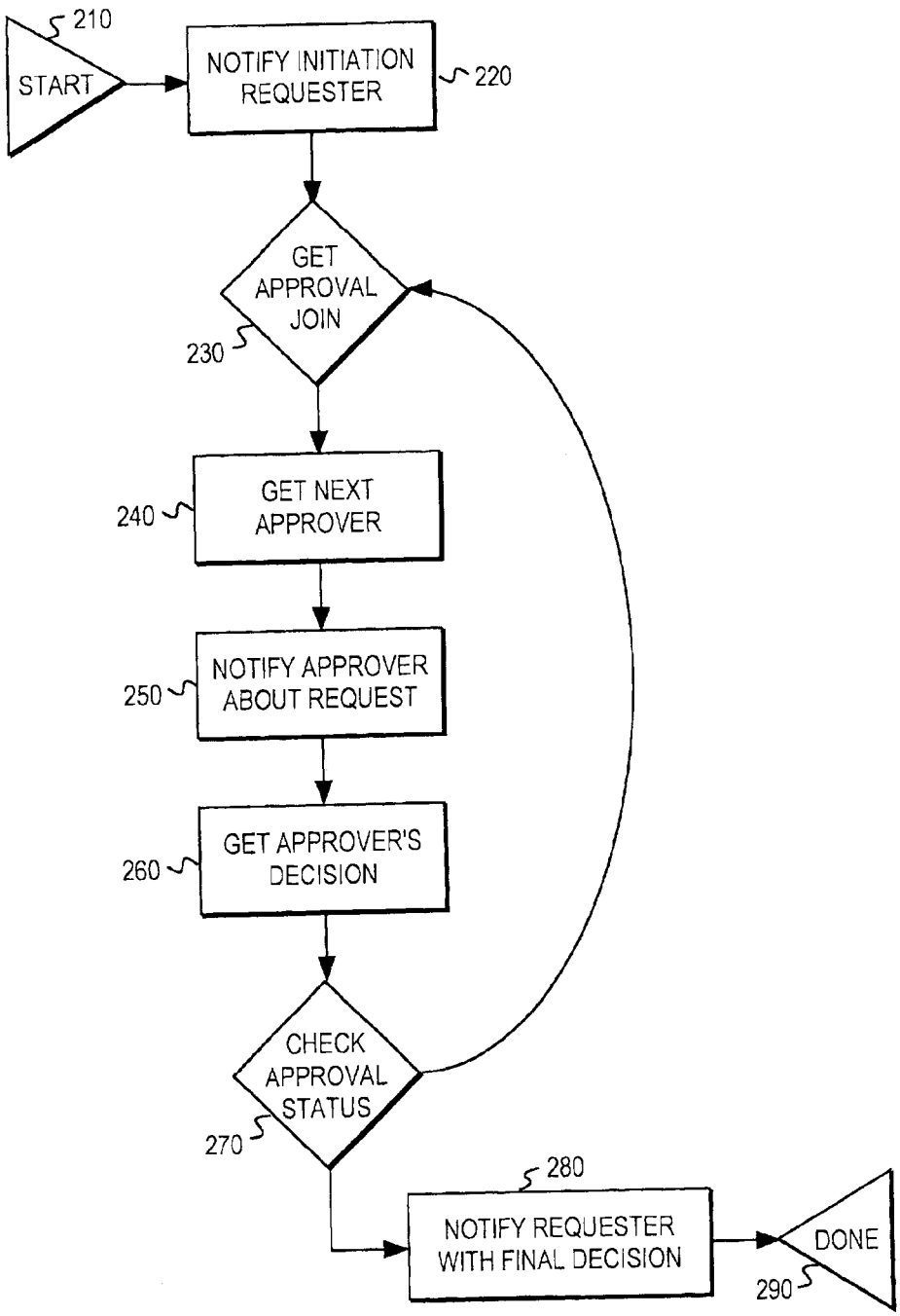

210
START
NOTIFY INITIATION REQUESTER
220
GET APPROVAL JOIN
230
GET NEXT APPROVER
240
NOTIFY APPROVER ABOUT REQUEST
250
GET APPROVER'S DECISION
260
CHECK APPROVAL STATUS
270
280
NOTIFY REQUESTER WITH FINAL DECISION
DONE
290

FIG. 2

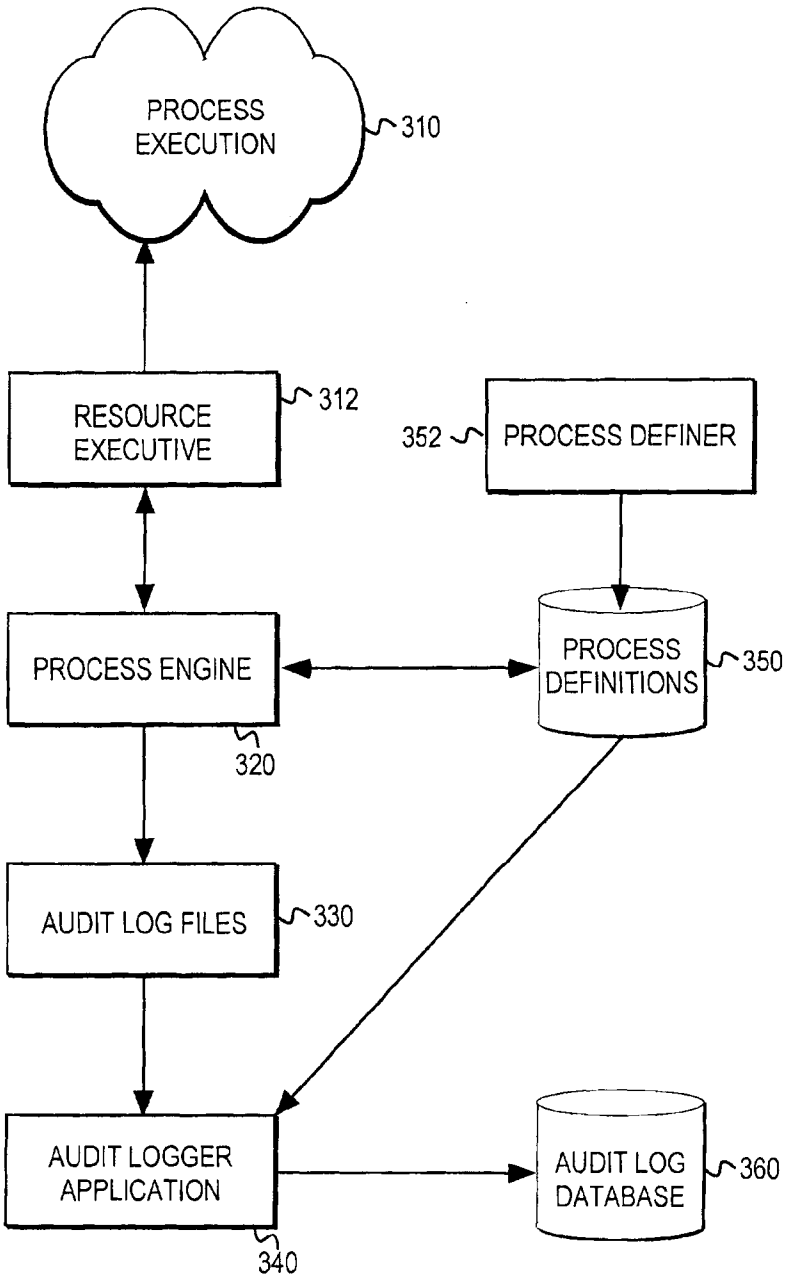
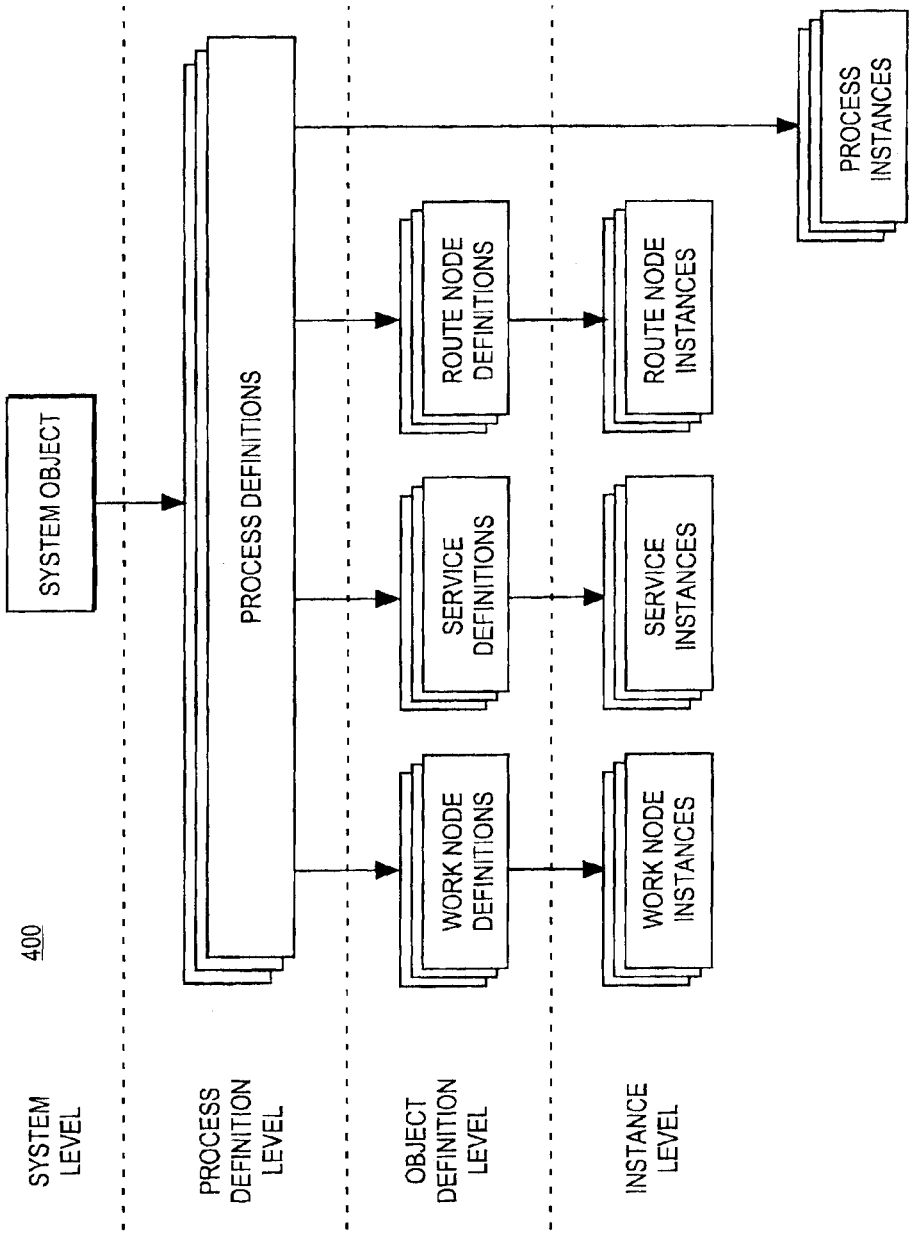

PROCESS EXECUTION
310
RESOURCE EXECUTIVE
312
352
PROCESS DEFINER
PROCESS ENGINE
320
PROCESS DEFINITIONS
350
AUDIT LOG FILES
330
AUDIT LOGGER APPLICATION
340
AUDIT LOG DATABASE
360

FIG. 3

FIG. 4


400
SYSTEM OBJECT
PROCESS DEFINITIONS
ROUTE NODE DEFINITIONS
SERVICE DEFINITIONS
WORK NODE DEFINITIONS
ROUTE NODE INSTANCES
SERVICE INSTANCES
WORK NODE INSTANCES
PROCESS INSTANCES
SYSTEM LEVEL
PROCESS DEFINITION LEVEL
OBJECT DEFINITION LEVEL
INSTANCE LEVEL

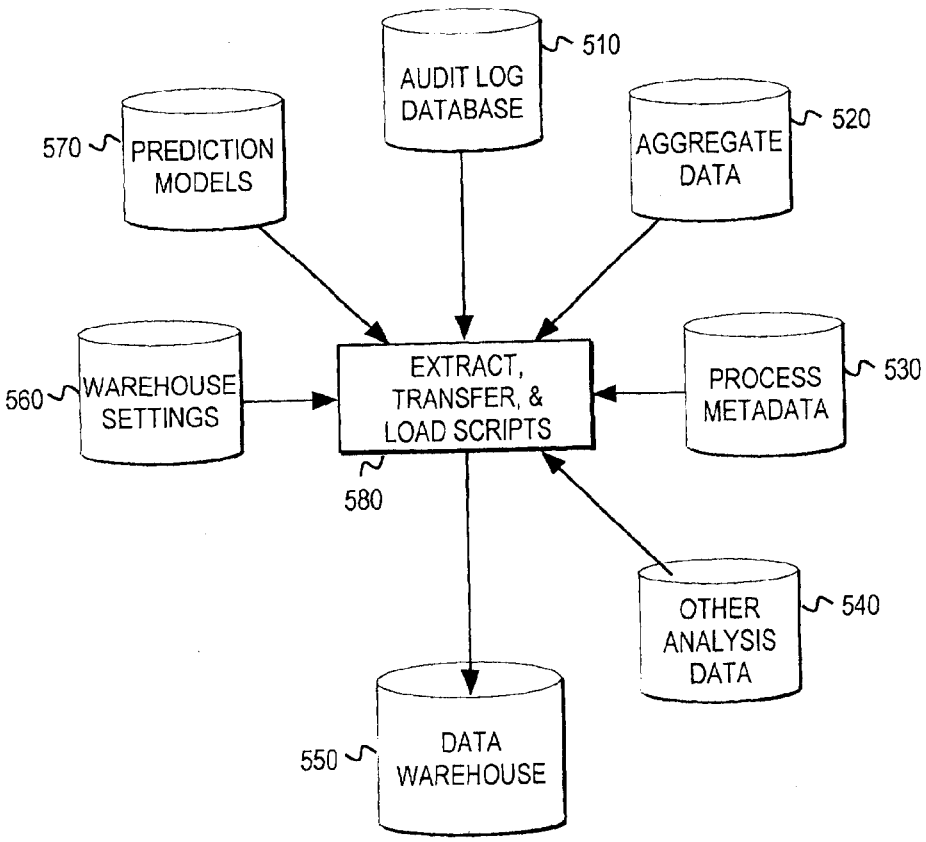

AUDIT LOG DATABASE
510
PREDICTION MODELS
570
AGGREGATE DATA
520
WAREHOUSE SETTINGS
560
EXTRACT, TRANSFER, & LOAD SCRIPTS
580
PROCESS METADATA
530
OTHER ANALYSIS DATA
540
DATA WAREHOUSE
550

FIG. 5

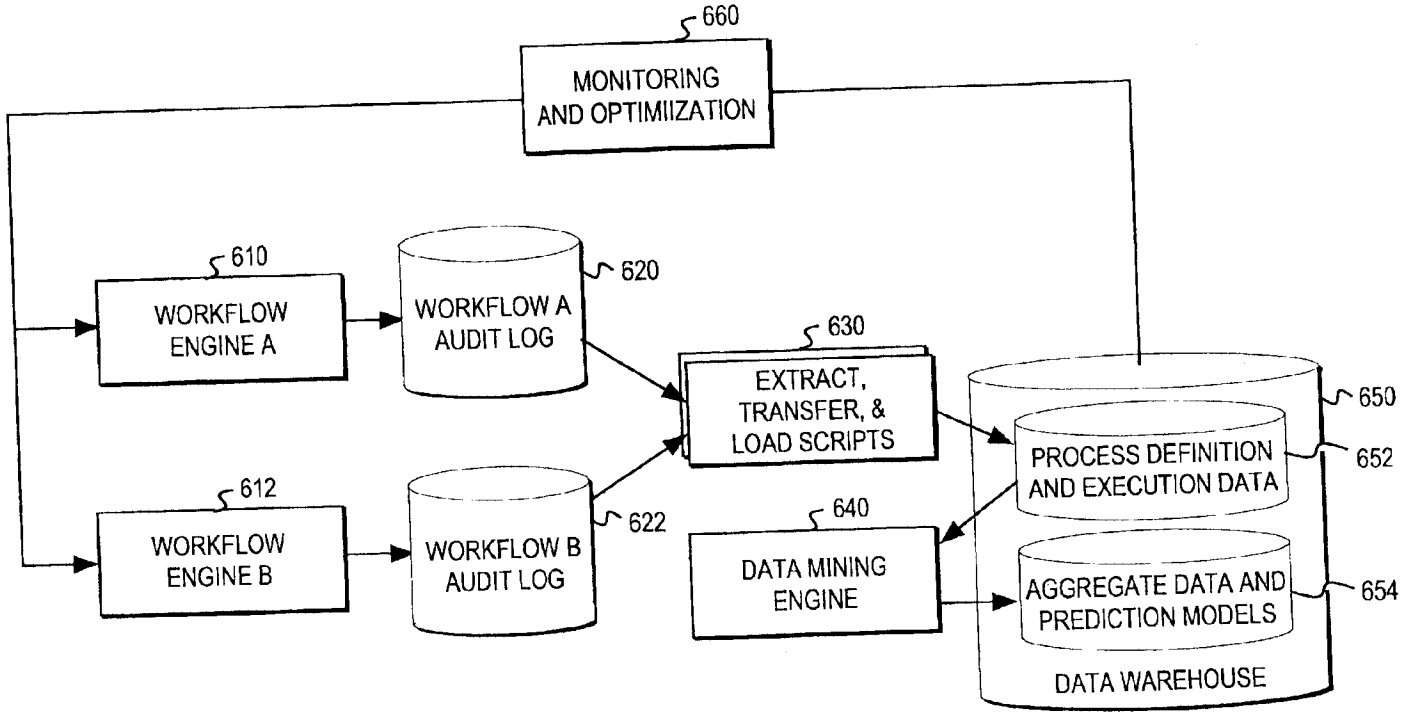

660
MONITORING AND OPTIMIIZATION
610
WORKFLOW ENGINE A
620
WORKFLOW A AUDIT LOG
630
EXTRACT, TRANSFER, & LOAD SCRIPTS
650
PROCESS DEFINITION AND EXECUTION DATA
652
612
WORKFLOW ENGINE B
622
WORKFLOW B AUDIT LOG
640
DATA MINING ENGINE
AGGREGATE DATA AND PREDICTION MODELS
654
DATA WAREHOUSE

FIG. 6

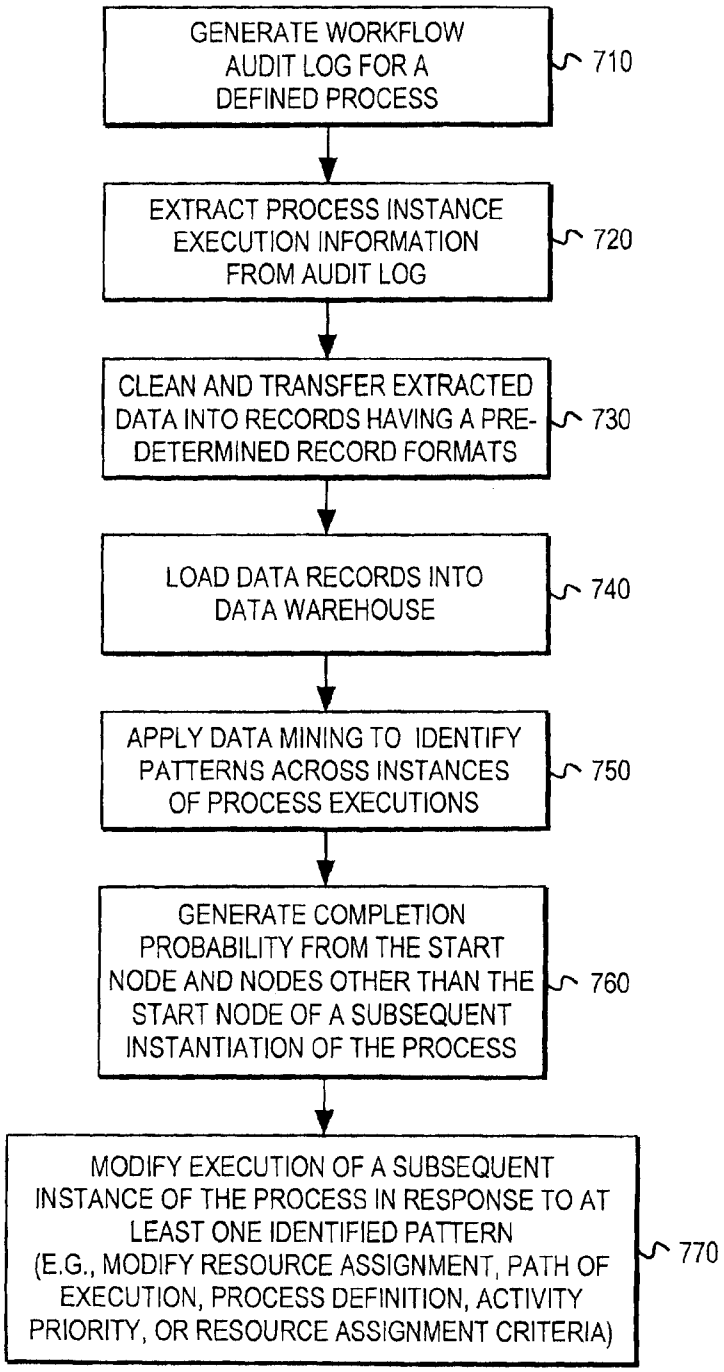

GENERATE WORKFLOW AUDIT LOG FOR A DEFINED PROCESS
710
EXTRACT PROCESS INSTANCE EXECUTION INFORMATION FROM AUDIT LOG
720
CLEAN AND TRANSFER EXTRACTED DATA INTO RECORDS HAVING A PRE-DETERMINED RECORD FORMATS
730
LOAD DATA RECORDS INTO DATA WAREHOUSE
740
APPLY DATA MINING TO IDENTIFY PATTERNS ACROSS INSTANCES OF PROCESS EXECUTIONS
750
GENERATE COMPLETION PROBABILITY FROM THE START NODE AND NODES OTHER THAN THE START NODE OF A SUBSEQUENT INSTANTIATION OF THE PROCESS
760
MODIFY EXECUTION OF A SUBSEQUENT INSTANCE OF THE PROCESS IN RESPONSE TO AT LEAST ONE IDENTIFIED PATTERN (E.G., MODIFY RESOURCE ASSIGNMENT, PATH OF EXECUTION, PROCESS DEFINITION, ACTIVITY PRIORITY, OR RESOURCE ASSIGNMENT CRITERIA)
770

FIG. 7

METHOD OF IDENTIFYING AND ANALYZING BUSINESS PROCESSES FROM WORKFLOW AUDIT LOGS

FIELD OF THE INVENTION

[0001] This invention relates to the field of business processes analysis, prediction, and optimization using computer generated workflow audit logs.

BACKGROUND OF THE INVENTION

[0002] Workflow management systems are used to monitor an organization's various administrative and production processes. These processes are defined in terms of activities, resources, and input and output process data. For a given process instance, the workflow management system might record information about the activities performed, when these activities are performed, time used to perform the activity, the identity of any resources involved in the activities, the outcome, and other data related to execution of the activities. This information is recorded as log data to permit subsequent reporting. Through various reporting tools the information is summarized and provided to analysts, workflow design, system administrator or other entities.

[0003] Typical workflow management systems permit users to query the execution state of a running process, report the number of process instances started or completed within a given time period, or compute simple statistics about groups of instances of a given process.

[0004] One disadvantage of traditional workflow management systems is a limited ability to address individual instance information both individually and relative to a collection or aggregate of instances.

[0005] For example, some workflow management systems place specific codes in data fields in the event of failure (e.g., "Jan. 1, 1970"). This data, however, invalidates aggregate calculations such as average activity execution time. In addition, queries that ensure proper calculation of aggregate values can be exceedingly complex to write. For example, writing queries that determine, for each fiscal quarter, the number of instances started and completed, the failure rate, and other quality/performance merits is difficult, time-consuming, and requires considerable database and workflow skills. As a result, traditional workflow management systems only offer very limited analysis functionality. In addition, they cannot make predictions about specific instances of a process or tune the process to improve process execution quality.

SUMMARY OF THE INVENTION

[0006] In view of limitations of known systems and methods, a method of identifying and analyzing business processes includes the step of populating a data warehouse database with data from a plurality of sources including an audit log, wherein the audit log stores information from a plurality of instantiations of a defined process. The data is then analyzed to predict an outcome of a subsequent instance of the process. Data mining techniques are applied to the data warehouse data to identify specific patterns of execution. Once the patterns have been identified, the outcome of a subsequent instance of the process can be predicted at nodes other than just the start node. The probability of completion information can be used to modify resource assignments in subsequent invocations of the defined process.

[0007] Other features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0009] **FIG. 1** illustrates an embodiment of a product manufacturing process.

[0010] **FIG. 2** illustrates one embodiment of an expense approval process.

[0011] **FIG. 3** illustrates a process definition and event logging system.

[0012] **FIG. 4** illustrates types of entities used for process definition.

[0013] **FIG. 5** illustrates a method of generating a data warehouse for one or more processes.

[0014] **FIG. 6** illustrates creation of a data warehouse for business processes.

[0015] **FIG. 7** illustrates one method of using workflow management audit logs to analyze and model business processes in order to predict and modify future behavior.

DETAILED DESCRIPTION

[0016] Processes may be modeled as a directed graph having at least four types of nodes including work nodes, route nodes, start nodes, and completion nodes. A process definition can be instantiated several times and multiple instances may be concurrently active. Activity executions can access and modify data included in a case packet. Each process instance has a local copy of the case packet. **FIG. 1** illustrates one embodiment of a process definition.

[0017] Node **110** represents a start node. The start node defines the entry point to the process. Each hierarchical definition level has at least one start node.

[0018] Nodes **120**, **130**, and **132** are examples of work nodes. A work node represents the invocation of a service or activity. Each work node is associated with a service description that defines the logic for selecting a resource or resource group to be invoked for executing the work. The service definition also identifies the case packet data items to be passed to the resource upon invocation (e.g., execution parameters or input data) and to be received from the resource upon completion of the work (e.g., status values, output data). Several work nodes can be associated to the same service description.

[0019] A service may be composed of a single atomic activity to be executed by a human or automated resource. Alternatively, a directed graph composed of a combination of work nodes and decisions may be referred to as a service. In this case, a service is analogous to a procedure or subroutine in an application program. The term "service" permits a convenient reference by name to a specific graph

of activities and decisions without re-iterating these individual components each time. For convenience sake, the series of activities may be invoked by referring to the service instead of the component sequence of tasks each time. The introduction of services enables a single definition to be re-used multiple times within the same process or in multiple processes. Thus a service may be used multiple times by a given process or by more than one process.

[0020] Node **140** represents a route or decision node. Route nodes are decision points that control the execution flow among nodes based on a routing rule.

[0021] Nodes **112** and **140** also control execution flow. Node **112** represents a fork in the execution flow. The branches may continue concurrently. Node **120** represents a joining of branches into a single flow. No further flow execution occurs until each branch preceding the join has completed. Join nodes and fork nodes are really special types of decision nodes.

[0022] Node **190** is a completion node. A process may have more than one completion node at a given hierarchical level.

[0023] **FIG. 2** illustrates a model of a business process for approving an expense. The process begins in start node **210** with the requester. The case packet data for the process might include the identity of the requester, the expense amount, the reasons, and the names of the individuals that should evaluate the request. Once the process is initiated, the requester is notified in work node **220**.

[0024] Work node **220** may invoke another service for notification. For example, notification might be performed by the service send_email. Upon invocation of the service, an email is sent to the requester notifying him that the process has begun. The process loops among the list of individuals until either all of them approves the expense or one of them rejects the expense (nodes **230-270**). (Join **230** is an OR join that fires whenever any input fires. The result is provided to the requester as illustrated by work node **280** before completion of the process at completion node **290**.

[0025] A workflow management system may be used to log execution data for different instantiations of a defined process. **FIG. 3** illustrates one embodiment of the use of a workflow engine to generate audit logs containing status information about different instantiations of one or more defined processes. Elements **352, 350, 312, 320, 330** and **340** may be collectively referred to as a workflow engine which generates an audit log database **360** containing information about process execution **310**.

[0026] Process definer **352** defines processes as a collection of nodes, services, and input and output parameters. These process definitions are stored in database **350**. The database may contain, for example, a process definition including a start node, a completion node, work nodes, route nodes, and services that the process is composed of. The process definition will also indicate how the nodes are connected to each other. The process definer **352** is used to specify the process definitions for the process definitions database **350**.

[0027] The process engine **320** executes processes by scheduling nodes to be activated. When a work node is activated, the process engine retrieves the associated service definition and resource assignment rule. The resource rule is communicated to resource executive **312**. The resource executive identifies the specific resources that should execute the service.

[0028] For example, the resource executive **312** selects specific resources such as a specific vendor, a specific employee, a specific piece of equipment, etc. The process engine controls the execution of processes. When executing a process, the process engine steps through the process definition to determine which activity should be performed next, and uses the resource executive **312** to assign a resource (or resources) to the activity. The process engine **320** then sends an activity along with the data required to perform the activity to the resource identified by the resource executive **312**. When the activity is completed, the process engine refers to the process definition to determine what happens next.

[0029] In one embodiment, the process execution information is written directly to an audit log database **360**. Alternatively, the process execution information is first written to audit log files **330** which serve as a buffer so that database performance does not adversely impact the recording function. The audit logger application **340** receives process definition information from database **350** and execution status information from the audit log files **330**. Audit logger application **340** stores at least a subset of the information in the audit log files into audit log database **360**. The user may choose to record different levels of information depending upon the purpose of the audit log. In one embodiment, databases **350** and **360** support an Open Database Connectivity (ODBC) application programming interface. The use of a buffer prevents database performance from impacting process execution. In particular, events that trigger a logging operation are not lost in the event the audit logger is unable to keep up with the process engine. The use of a buffer also enables updates to database **360** to be organized for efficiency rather than being driven directly by events as they occur in the executing process.

[0030] The audit logger **340** uses the events recorded in the audit log files **330** and the definitions from the process engine database **350** to generate various statistics about process events or to log information on individual processes. The information generated by the audit logger application **340** is stored in the audit logger database **360**. The amount of information logged for each process instance varies depending upon the level of logging defined for the process.

[0031] The audit log database provides information regarding particular instances of a process. For example, a particular instance may be identified by a unique identifier, the start time and the completion time of the process instance. Node instance information describes an element or step such as a work node or a route node in a process definition. Exemplary node information includes a unique node identifier, the time the instance of the node was created, and the time the instance of the node was completed. Activity instance information describes the activity or set of activities generated by a work node. The type of activity, time the activity instance was created, and the time the activity instance was completed are examples of information that may be logged for activities.

[0032] **FIG. 4** illustrates a hierarchy **400** for entities about which information may be reported from audit log database

360. With respect to processes, the user may select to have only the identity of defined processes logged (i.e., process definition level). If more detail is required, the user may elect to have work node definitions, service definitions, and route node definitions for each defined process logged (i.e., object definition level). If still more detail is desired, information about each instantiations of work nodes, services, and route nodes may be recorded (i.e., instance level).

[**0033**] Depending upon the level of reporting desired, the information stored within the audit log database may include process identity, start date/time, completion date/time, start and completion date/time for each work node, specific resource assignments for work nodes, input and output data or parameters for each work node, etc.

[**0034**] Data mining techniques such as pattern matching and classification are then applied to the contents of the data warehouse including the audit logs to identify patterns occurring during process execution. These patterns may be used to predict process execution quality, workload on the system and on the resource, and more. For example, the patterns may be used to predict the completion of subsequent instances of the process from nodes other than a start node. Data mining uses pattern recognition, statistical, and other mathematical techniques to identify correlations, patterns, and trends. Large amounts of data may be selected, explored, and modeled with pattern matchers, for example, to identify specific conditions under which exceptions or significant changes in performance occur.

[**0035**] Analyzing the workflow warehouse with data mining techniques can reveal that a specific resource fails or is incapable of meeting process requirements under certain conditions which are not otherwise obvious to the observer and may in fact be inter-related with conditions seemingly unrelated to the resource. Generally, these techniques may identify conditions for which process execution quality departs from typical or average quality or is incapable of meeting a service level agreement. The user must select a sufficient level of reporting detail to ensure that data directly related to the cause or correlated with the cause of these differences in performance are stored in the audit logs.

[**0036**] For example, if one machine is not performing properly, the audit log database and the warehouse must have resource assignment information to identify the problem (causation). If throughput improves at different times of day or on different days of the week, for example, due to the availability of better performing resources, then recordation of the start and stop times rather than just elapsed time will at least enable the discovery of information highly correlated with the cause even if specific resource assignments are not recorded. The pattern information enables analyzing the process or processes so that predictions may be made with respect to subsequent process instantiations that match the pattern. The pattern information enables the derivation of rules to describe the behavior. The rules, in turn, are the basis for subsequent analysis and the predictive models. The rules may be examined to determine the cause or at least identify events highly correlated with the cause.

[**0037**] In order to identify patterns and make predictions, specific process instance information as well as aggregate information about the status of process instance executions are required. This information is collected and stored in a data warehouse for analysis along with other data necessary for generating the type of information and in a format desired by the user.

[**0038**] **FIG. 5** illustrates the types of data that may be used for analysis. The audit log database **510**, aggregate data **520**, process metadata **530** (e.g., process properties including cost, priority, etc.), prediction models **570**, warehouse settings **560**, and other analysis data **540** are loaded into data warehouse **550**. The data warehouse may also contain the definitions of processes, nodes, or resources that can be associated with behavior of interest. Extract, transfer, and load scripts **580** may be used to obtain the audit log **510**, warehouse setting **560**, and process metadata **530** information for the data warehouse.

[**0039**] The audit log database **510** is generated by the workflow engine. The aggregate database may be generated by other applications such as the data mining application. The aggregate database may include averages, counts, maximum, minimum, etc. values for various monitored process execution data. The aggregate data is calculated from historical execution data and continuously updated as subsequent instances of the process are invoked.

[**0040**] The prediction models are generated and updated by the data mining process. The warehouse settings and other analysis data are provided by the user. The warehouse settings typically includes control settings for the data warehouse and other information related to maintenance of the data warehouse. The other analysis data may include trend lines or models that the user desires to compare the process execution performance with that is distinct from the aggregate data.

[**0041**] In one embodiment, the data warehouse provides a structured query language (SQL) interface for accessing and maintaining the data. Thus standard commercial reporting tools can still be used to generate reports.

[**0042**] Some of the extract, transfer, and load (ETL) scripts are tailored for the specifics of the source database. Thus, for example, in the presence of audit logs produce by workflow management applications from different vendors, the ETL scripts must include scripts tailored to accommodate the vendor-specific source record format and idiosyncrasies with respect to data values. The ETL scripts must extract the data from the audit logs. The extracted data must then be normalized. If, for example, start and stop times are recorded in different formats for audit logs from different vendors, the time values are converted to a common format. The data must also be "cleaned" to ensure that vendor-specific audit mechanisms do not impair the ability to properly calculate aggregate values. In particular, the use of default values in fields used for aggregate calculations are avoided.

[**0043**] For example, elapsed execution times may be pre-calculated for storage by the audit logger. Alternatively, elapsed execution times may subsequently calculated by subtracting the start times from the stop times. The use of default date/time values for stop time in the event of process exceptions would result in an invalid elapsed time, which in turn would adversely affect aggregate calculations (e.g., averages). The ETL script for a specific audit logger must be aware of vendor-specific implementations in order to properly clean the data for subsequent processing. Instead of a

default date/time value, for example, a null value may be used so that aggregate elapsed time calculations would not be affected. Once the data has been cleaned and transferred into a common format from possibly different vendor formats, the data is loaded into the data warehouse.

[0044] **FIG. 6** illustrates the path of data flow for identifying and analyzing business processes. The method can be applied to processes being tracked by multiple workflow engines **610, 612** which may be from different vendors. Each workflow engine **610, 612** generates a corresponding audit log **620, 622**. The extract, transfer, and load scripts **630** are applied to populate the data warehouse with process definition and instance execution data **652**. Some of the extract, transfer, and load scripts **630** are specifically designed to accommodate their corresponding vendor-specific audit logs **620** and **622**. The ETL scripts also generate some aggregate information. Other aggregate data is specified in terms of views and therefore maintained and updated by the database.

[0045] Data mining engine **640** operates on the process definition and execution data **652** to generate aggregate data and prediction models **654**. Based on patterns identified from data mining analysis, the prediction models, for example, can reveal rules that can be applied to running process instances to predict their outcome, completion time, the services and resources involved in the execution, etc. The use of aggregate data alone would not otherwise take into account patterns that occur with respect to specific resource assignments.

[0046] The prediction models may then be used by monitoring and optimization block **660** to modify resource assignments for subsequent process instances and to make other optimizations by changing process and system characteristics. In one embodiment, the prediction models may be used to identify the risk of an undesirable pattern and then re-assign resource assignments to prevent realization of the undesirable pattern. Alternatively, the monitoring and optimization block **660** may update the workflow engines to re-prioritize resource assignments, modify resource assignment criteria, or modify process definitions in order to reduce the likelihood of the realization of an undesirable pattern.

[0047] **FIG. 7** illustrates one embodiment of a method for identifying and analyzing business processes from a workflow audit log. In step **710**, a workflow audit log is generated for instances of execution of a defined process. In step **720**, the desired process instance execution information is extracted from the audit log. The extracted data is cleaned and transferred into records with pre-determined formats in step **730**. This ensures data from different vendor audit logs can be put into a common format for subsequent analysis. The data records are then loaded into the data warehouse in step **740**. Steps **720-740** are handled by extract, transfer, and load scripts in one embodiment.

[0048] In step **750**, data mining is applied to the data warehouse data in order to identify patterns across instances of process executions. Data mining enables 1) discovery of the actual business process followed in the organization and modifications of the defined workflows to better match these business processes; 2) understanding the performance and quality both in general or relative to other resources or with respect to the execution of specific services, nodes, or processes; 3) identifying the causes of behaviors of interest such as process execution characterized by a very high or low quality; 4) derivation of rules and prediction models that can be used to make predictions for process execution outcome, duration, invoked services, invoked resources, system load, and resource load; and 5) tracking, monitoring, and reporting of process metrics.

[0049] For example, the resources can be rated relative to other resources depending on the work they perform and when the work is performed. The prediction models may be used to predict whether a node will be activated or not and if so then how many times. Similarly, the prediction models may be used to predict the use of a resource and the load on the system and the resources. The prediction models may be used on executing process instances to modify routing rules, resource assignment, or other characteristics dynamically, for example, to improve process throughput or process execution quality. For example, the prediction models may be used to dynamically modify any of 1) a selection of resources applied to individual activities of the process; 2) a path of execution; 3) a process definition; 4) an activity priority, and 5) a resource assignment criteria for the subsequent instance of the process in response to a result of the analyzed data.

[0050] In step **760**, completion probabilities from the start node and nodes other than the start node can be generated for subsequent instantiations of the process. In step **770**, execution of a subsequent instance of the process is modified in response to at least one identified pattern. As discussed above, the process may be dynamically modified by performing any of the steps of modifying the resource assignment, modifying the execution path, redefining the process, changing the activity priority, or changing the resource assignment criteria.

[0051] In the preceding detailed description, the invention is described with reference to specific exemplary embodiments thereof. Various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A method comprising the steps of:

a) populating a data warehouse database with data from a plurality of sources including an audit log, wherein the audit log stores information from a plurality of instantiations of a defined process;

b) analyzing the data to predict an outcome of a subsequent instance of the process.

2. The method of claim 1 further comprising the step of:

c) modifying at least one of a selection of resources applied to individual activities of the process, a path of execution, a process definition, an activity priority, and a resource assignment criteria for the subsequent instance of the process in response to a result of the analyzed data.

3. The method of claim 1 wherein step b) further comprises the step of predicting the outcome at a plurality of nodes within the defined process.

4. The method of claim 1 wherein step b) further comprises the step of:

applying a pattern matcher to the data to identify patterns of execution.

5. The method of claim 1 wherein step b) further comprises the step of:

applying data mining techniques to the data warehouse to identify patterns of execution.

6. The method of claim 1 further comprising the step of:

c) modifying a selection of resources applied to individual activities of the process in response to the predicted outcome.

7. The method of claim 1 further comprising the step of:

c) modifying a selection of an execution path within the process in response to the predicted outcome.

8. The method of claim 1 further comprising the step of:

c) modifying a priority of the process in response to the predicted outcome.

9. The method of claim 1 further comprising the step of:

c) analyzing the data to identify patterns corresponding to a cause of at least one of a selected predicted outcome and a selected actual outcome.

10. The method of claim 1 further comprising the step of:

c) analyzing the data to identify patterns corresponding to a high correlation with a cause of one of a selected predicted outcome and a selected actual outcome.

11. The method of claim 1 further comprising the step of:

c) analyzing the data to identify patterns resulting in outcomes representing a departure from an average outcome for at least one measured process metric.

12. A method comprising the steps of:

a) populating a data warehouse database with data from a plurality of sources including an audit log, wherein the audit log stores information from a plurality of instantiations of a defined process;

b) analyzing the data to identify process outcome classification rules; and

c) predicting completion probability from at least one node other than a start node of a subsequent instantiation of the defined process.

13. The method of claim 12 further comprising the step of:

d) modifying at least one of a selection of resources applied to individual activities of the process, a path of execution, a process definition, an activity priority, and a resource assignment criteria for the subsequent instantiation of the process in response to at least one of the predicted completion probabilities.

14. The method of claim 12 wherein step b) further comprises the step of predicting the completion probability at a plurality of nodes within the defined process.

15. The method of claim 12 wherein step b) further comprises the step of:

applying a pattern matcher to the data to identify patterns of execution.

16. The method of claim 12 wherein step b) further comprises the step of:

applying data mining techniques to the data warehouse to identify patterns of execution.

17. The method of claim 12 further comprising the step of:

d) modifying a selection of resources applied to individual activities of the process in response to at least one of the predicted completion probabilities.

18. The method of claim 12 further comprising the step of:

c) modifying a selection of an execution path within the process in response to at least one of the predicted completion probabilities.

19. The method of claim 12 further comprising the step of:

c) modifying a priority of the process in response to at least one of the predicted completion probabilities.

20. The method of claim 12 further comprising the step of:

c) analyzing the data to identify patterns correlated with selected completion probabilities.

* * * * *

UNITED STATES DISTRICT COURT
MIDDLE DISTRICT OF FLORIDA
OFFICE OF THE CLERK
UNITED STATES COURTHOUSE
TAMPA, FLORIDA 33602

OFFICIAL BUSINESS